View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Carolina Digital Repository

DATA SCIENCE METHODS WITH APPLICATIONS TO GENETIC SEQUENCING

Weiwei Li

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill
2020

Approved by:

Jan Hannig

Corbin D. Jones

Sayan Mukherjee

Kai Zhang

Nicolas Fraiman

© 2020
Weiwei Li
ALL RIGHTS RESERVED

ABSTRACT

WEIWEI LI: Data Science Methods with Applications to Genetic Sequencing
(Under the direction of Jan Hannig, Corbin D. Jones, and Sayan Mukherjee)

Data science methods are of increasing importance in modern genetic sequencing analysis. In this dissertation, we focus on applying statistical modeling to structural variant detection problems and a new framework for scalable and provable subspace clustering.

We first discuss the optimal sampling strategy for structural variant detection using optical mapping. Here we develop an optimization approach using a simple, yet a realistic, model of the genomic mapping process using a Hypergeometric distribution and probabilistic concentration inequalities. Surprisingly we show that if a genomic mapping technology can sample most of the chromosomal fragments within a sample, comparatively little biological material is needed to detect a variant at high confidence.

In the second part, we introduce a formal probabilistic model to assessing how well an optical maps to a reference genome. We use this approach to infer the most likely location within that reference for any given read, as well as the likelihood of mapping to all other possible locations. Using data produced by BioNano Saphyr to parameterize a simulation, we show that our approach accurately identifies the likeliest locations of the observed optical read data. While considerably faster than a canonical MCMC approach, our approach is still computationally intensive. We provide several algorithmic improvements that increase the speed with no apparent impact on accuracy. Our approach provides a rigorous, open-source framework for analyzing optical read data.

In the third part, we introduce a scalable and provable algorithm for subspace clustering. Specifically, we consider modeling the collection of points in a high dimensional ambient space as a union of lower-dimensional subspaces. In particular, we propose a scalable sampling-based algorithm that clusters the entire dataset via first spectral clustering of a small random sample followed by classifying or labeling the remaining out of sample points. The key idea is that this random

subset borrows information across the entire dataset and that the problem of clustering points can be replaced with the problem of "clustering sub-clusters". We provide theoretical guarantees for our procedure. The numerical results indicate that we outperform other state-of-the-art subspace clustering algorithms with respect to both accuracy and speed.

To my parents

ACKNOWLEDGEMENTS

It is a truth universally acknowledged, that a Ph.D. student in possession of a great graduate study, must be surrounded by many great people. Now my school life is coming to an end, and I would like to express my sincere gratitude to all the people who have contributed a lot to my awesome five years at Chapel Hill.

First and foremost, I am indebted to my advisors, Prof. Jan Hannig, Prof. Corbin D. Jones and Prof. Sayan Mukherjee. Thank you so much for your guidance and supports during my graduate study. Jan, I truly appreciate that you granted me the flexibility to expand my research interests and do internships in industry. I have learned a lot from our technical discussions and am constantly amazed by your innovative ideas. Corbin, your optimism personality makes my whole research experience enjoyable. It is inspiring to see an established professor like you still so humble and eager to learn new things. Sayan, thank you so much for accepting me as your student, I know it is very rare in Duke. I also feel grateful that you have broadened my vision in research and motivated me to explore more in the intersection of traditional statistics and computer science. In retrospect, I am so lucky to have you three as my advisors. Finishing graduate study is just another starting point of my new journey, and I believe one day, you will be very proud of me.

Besides my advisors, I would like to thank the rest of my dissertation committee members Professor Kai Zhang and Professor Nicolas Fraiman. Kai, I really enjoyed your course in generalized linear models, that course motivated me to think about linear models more from a plane perspective. Nicolas, I know that you have a very busy schedule this semester and thank you so much for taking the time to be my committee member.

I am also thankful to my dear friend, Haokun, until these days I am still motivated by his words: "Ph.D. study is not about a certain set of technical skills, it is about gaining the ability to solve mentally and intellectually challenging problems." I also extend my gratitude to my great buddies Ming, Zhengyang and Shiwei, for their constant companionship–studying with me, training with me, laughing with me, and inevitably, being foolish and crazy with me. There were some dark days, and

I am so lucky to have you guys back me up throughout this journey. Last but not least, I would like to express my deepest gratitude to my dear parents for their ineffable love and unconditional support. My mom has always been my role model and mentor since my early childhood. She deserves my warmest thanks for her careful guidance through each stage of my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Large and small structural variation within the genome contributes to phenotypic diversity among individuals and impact human health. Advances in genome sequencing have given new insight into some of this variation, but accurate and through characterization remains elusive. Emerging optical mapping technologies, such as the BioNano Genomics platform and similar tools, can provide high resolution characterization of structural variation. As these technologies are expected to evolve into clinical diagnostic tools, it is critical to develop robust statistical methods for assessing the credibility of the structural variation discovered. In the second and third chapters of this dissertation, we try to develop data science methods that can be used to analyze the structural variants data.

In the era of "Big Data", tremendous data points are collected from different sources. Data scientists nowadays need to analyze datasets with large volume and high-dimensionality, which leads to an urgent need for scalable algorithms. In fact, the information contained in a dataset does not necessarily grow with the dimensionality. In many machine learning problems (motion segmentation, face recognition, image compression etc.), the data points usually lie in a union of low-dimensional linear subspaces. Finding these subspaces and assigning the cluster membership to each data point is the main goal of subspace clustering. In this dissertation, we develop a scalable and provable subspace clustering algorithm, dubbed Sampling Based Subspace Clustering (SBSC). Numerical experiments demonstrate that SBSC outperform other state-of-the-art algorithms in medium-sized and large-sized datasets.

1.1 Optimal Sampling for Variant Detection

In the second chapter, we want to design a sampling strategy that determines how many biological materials are needed in order to detect the structural variants with high confidence. Structural variants compose the majority of human genetic variation, but are difficult to accurately assess using current genomic sequencing technologies. Optical mapping technologies, which measure the size of chromosomal fragments between labeled markers, offer an alternative approach. As these technologies

mature towards becoming clinical tools, there is a need to develop an approach for determining the optimal strategy for sampling biological material in order to detect a structural variant at some threshold. Here we develop an optimization approach using a simple, yet realistic, model of the genomic mapping process using a Hypergeometric distribution and probabilistic concentration inequalities. Our approach is both computationally and analytically tractable and includes a novel approach to getting tail bounds of Hypergeometric distribution. We show that if a genomic mapping technology can sample most of the chromosomal fragments within a sample, comparatively little biological material is needed to detect a variant at high confidence.

The full process of optical mapping can be described as an urn sampling problem, which in turn can be statistically modeled as sampling from Hypergeometric distributed random variables. The tail bounds of Hypergeometric distribution was discussed in Skala [2013]. In this dissertation, we followed this path and extended it with a general result. While these bounds work pretty well if the probability of success p is near 0.5, in our particular application (i.e. optical sampling) p is usually very small, making the previous bounds very conservative. Therefore, we used the tail bounds from Binomial distribution as an approximation. These Binomial tail bounds are well studied and particularly, we used a bound from Short [2013] that is relatively tighter for small p.

Based on these tail bounds on Hypergeometric distribution, we designed a sampling strategy for optical mapping. The simulation study shows that our algorithm has similar behavior with concentration inequality free results.

1.2 Semi-Parametric Model for Optical Mapping

In optical mapping, the biological materials (i.e. genetics) are represented as vectors of positive integers. To detect structural variant, one needs to map the reads from optical mapping back to the reference genome. Mathematically speaking, the optical mapping device maps a bunch of sub-sequences (original sequences) from a long vector (reference genome) into new vectors (optical reads), all these vectors have positive integer entries. Given the reference genome and optical reads, we want to find the corresponding original sequences of these optical reads by modeling the mapping procedure as a function with random outputs.

Previous approaches usually apply a heuristic score function together with dynamic programming to find these original sequences [Anantharaman et al., 1997, Leung et al., 2017b]. While these approaches work fairly well on finding these original sequences with decent running time, they do

not have a probabilistic model to measure the likelihood of each candidate region and are usually sensitive to hyper-parameters set for score functions.

In this dissertation, we use semi-parametric regression to model the uncertainty during optical mapping. This semi-parametric approach is highly flexible with few hyper-parameters and can approximate a variety of random distributions. The maximum likelihood estimators of this semi-parametric model can be found by using a two-step projected gradient descent method. The whole framework of our algorithm includes two stages: (1) finding the potential original regions by a binary matrix filtering algorithm, which is highly scalable and much faster than canonical MCMC method; (2) calculating the likelihood for each candidate region and output the candidate sequences ranked by their corresponding likelihoods.

In this dissertation, we only test the performance of our procedure with binary matrix filtering algorithm on simplified synthetic datasets that do not have complicated mapping variations like deletions, insertions and trans-locations. The numerical result shows that our algorithm is much better at handling large uncertainty of sizing errors than a state-of-the-art alignment algorithm called OMBlast. It is also convenient to stack other filtering algorithms with the semi-parametric model and extend our algorithm to general usage.

1.3 Subspace Clustering through Sub-Clusters

In modern data analysis, researchers and practitioners often need to handle high-dimensional datasets with large data volume. Training machine learning models directly on these datasets can induce huge computational cost and hence are usually prohibitive. Therefore, dimension reduction is usually a desired pre-processing step [Hotelling, 1933].

In Chapter 4, we consider the subspace clustering problem [Elhamifar and Vidal, 2009]. In which we assume the data points from high-dimensional ambient space lie in a union of linear subspaces. Our goal is to find the membership of each data point with respect to these subspaces. In the downstream analysis after dimension reduction, one can easily run PCA [Hotelling, 1933] on each cluster to get its corresponding dimension and orthogonal base.

Previous works on subspace clustering are usually not scalable. In this dissertation, we developed a sampling-based subspace clustering framework, which runs very fast even in large datasets and has provable performance guarantee. The numerical experiments demonstrate a significant improvement of our algorithm over other state-of-the-art algorithms.

CHAPTER 2

Optimal Sampling for Variant Detection

2.1 Introduction

Structural variants (SV), insertions, deletions and trans-locations, are by far the most common types of human genetic variation [Chaisson et al., 2015]. They have been linked to large number of heritable disorders [Hurles et al., 2008]. Technologies to assay the presence or absence of these variants have steadily improved in ease and resolution [Huddleston and Eichler, 2016, Audano et al., 2019]. Whole genome shotgun DNA sequencing (WGS) can detect small variants (less than 10bp) readily and can detect some classes of large SV. This approach, however, is inferential and often struggles to capture copy number variation in gene families or to correctly estimate the size of insertions. An alternative approach, genomic mapping (such as the technology of BioNano Genomics), addresses the deficiencies of WGS by providing linkage and size information from ordered fragments of chromosomes spanning tens to hundreds of kilobases. In contrast to WGS, genomic mapping approaches directly observe SV, rather than inferring the existence of a SV from patterns of mismatch in WGS data. In the near future, these genome mapping technologies are expected to be used for clinical diagnosis of SV known to be associated with genetic disorders.

In a clinical setting, the cells or tissues needed for analysis may be hard to obtain, which poses several important statistical questions: what is the minimum amount of starting material necessary to have some confidence of detecting a target fragment? What is the optimal sampling strategy for the primary and derived material throughout the process? How best to model the technical errors–such as failure to digest at a site–during the processing of the data as these errors can lead to false positives and negatives? As is often the case, answering these questions motivated an exploration and expansion of the statistical machinery used to model this biological process.

Our contributions are twofold. From the algorithmic perspective, we explored, both theoretically and empirically, the connection between Hypergeometric distribution and binomial distribution. We showed that under certain conditions, the tail bounds of binomial distribution can be used to control

that of Hypergeometric distribution. A direct tail bound on hypergeometric is also developed in this dissertation. From the clinical and experimental perspective, we built an extensible model for estimating the amount of material needed for optical mapping of a genome. As these technologies move into clinical practice–such as diagnostics for chromosome abnormalities–there is critical need to be able to determine if enough genomic material is available for applying this assay.

The rest of this chapter is organized as follows: in Section 2.2, we present the problem from a biological perspective; in Section 2.3, we describe the statistical modeling of the sampling problem and our sampling strategy; in Section 2.4, we introduce the implementation details of our sampling algorithm; in Section 2.5 we present our numerical results on synthetic data sets; in Section 2.6 we summarize the conclusions of this chapter. Proofs are relegated to the appendix.

2.2 Approach

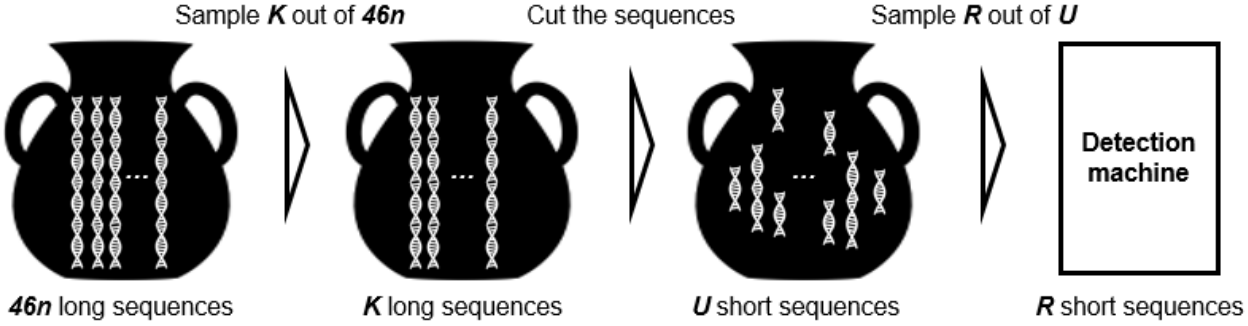
The starting input for genomics mapping technologies is often an aliquot of cells isolated from the tissue of interest. The technology then performs an "on-chip" digestion of these cells, followed by extraction of the nucleic acids from these cells. While efforts are made to maintain intact chromosomes, these long DNA molecules (50-250 million base pairs [bp] of DNA per chromosome in humans; "long sequence") often experience one or more breaks during extraction. These long sequences are then elongated either on a slide or in a nanochannel and probed for specific short DNA sequences within the the long sequences using either optical probes or restriction enzyme digest(nicking/cutting) based methods. These short sequences are usually short and found every 1000-100,000 bp. As the long sequence moves through the nanochannel each possible short sequence is evaluated beginning with the first bp. For a given sequence, this produces a list of lengths (bps) of distances between detected short sequences that is ordered by when they occurred in the initial long sequence. In the case where we are assaying for a specific SV (the *sequence*), these lists are compared to a reference list generated from the human reference genome from that particular chromosomal region. Discrepancies between these two lists in the target regions potentially indicates a SV. However, to be called a SV a minimum amount of evidence must be obtained supporting the variant. Typically, this threshold is 5 to 50 examples of the discrepancy in the observed data.

As the cells or tissues needed for analysis may be hard to obtain, these technologies pose several important statistical questions: what is the minimum amount of starting material necessary to have some confidence of detecting a target fragment? What is the optimal sampling strategy for the

Table 2.1: Nonrandom quantities

Notation	Definition
n	Number of cells in the first urn (copies of each type of long sequences).
K	Number of sequences sampled from the first urn.
R	Number of sequences sampled from third urn.
L	Approximated length of long sequence.
l	Approximated length of short sequence.
T	Threshold on detectability of target sequences.
f	Length of fragment of interest.
c	Approximated ratio between lengths of long and short sequences.
Q	Minimum number of target sequences we want in the detection machine.
p	Minimum confidence in achieving the goal.

Figure 2.1: Urn Demonstration of Sampling Procedure



Three urn demonstration of the algorithm: the first urn contains raw biological materials; the second urn contains materials sampled from the first urn; the third urn contains materials from the second urn that are cut into shorter segments. Content of the third urn is sampled and assayed in the detection machine.

primary and derived material throughout the process? How best to model the technical errors–such as failure to digest at a site–during the processing of the data as these errors can lead to false positives and negatives? Our goal here is to begin to address these issues by developing a straightforward statistical model and using this model to determine the optimal sampling strategy.

2.3 Statistical Model

In this section, we abstract our sampling procedure into an “urn sampling" model. As DNA is processed through the optical mapping procedure, we imagine the material passing through a series of urns. Assume we have 46 different types of long sequences (i.e. chromosomes), each type has n copies (i.e. n cells), so we have $46n$ long sequences in total. We assume only one type of long sequences contains the target sequence, or the fragment of interest. The basic idea of our sampling model is shown in Figure 1. The notations introduced below are summarized in Table 2.1.

The first urn contains our original biological material, total of $46n$ long sequences out of which n of them contain the target sequence. At the first stage, we sample K sequences without replacement from the first urn, and put them in the second urn. The second urn will therefore contain a random number X of target sequences. All of the K long sequences in second urn are cut (a.k.a. nicked and labelled) at random locations according to a Poisson process and placed into the third urn. The third urn will therefore contain a random number of U sequences out of which W are target sequences. The content of the third urn models the biological material prepared for assay in a detection machine. Finally, we sample R smaller sequences without replacement out of the third urn and put them into a detection machine. There will be a random number Y of target sequences processed by the detection machine, and the goal is to assure that for some pre-specified values Q and p, we have the probability of $Y \geq Q$ is at least p. Throughout the experiment, the variables (n, K, R) are in our control and we will find the conditions on them to achieve our goal. In this chapter, we call the long sequence in the second urn which contains the fragment of interest as "target sequence".

Next we state the following biological assumptions for our modeling

1. The length of target sequence is f.

2. The lengths of long sequences in the first urn are approximately L, here $L \gg max(f, T)$.

3. Short sequences in the third urn have lengths approximately l, and $c \approx \frac{L}{l}$.

We proceed by describing the probabilistic parts of our model. The distributions and their expectations are summarized in Table 2.2. There are X target sequences in the second urn. It is straightforward to see $X \sim H(46n, n, K)$, a Hypergeometric distribution with $46n$ samples, n samples of interest and K as sampling size.

Let U_i $(i = 1, 2, .., K)$ denotes the number of cuts on i-th long sequence in the second urn. Combine with the third assumption above, we assume that U_i follows a Poisson distribution with mean c. Note that U_i cuts divide the sequence into $(U_i + 1)$ shorter sub-sequences. Consequently, $U = \sum_{i=1}^{K}(U_i + 1)$ is the total number of short sequences in the third urn, and $(U - K)$ follows Poisson distribution with mean cK.

Write W as the number of the sequences in the third urn that contain the target sequence. The distribution of W is more complicated than that of X. Assuming $X > 0$, we have at least 1 target

Table 2.2: Random quantities and their expectations

Notation	Distribution	Expectation
X	$H(46n, n, K)$	$\frac{K}{46}$
U_i	$Poi(c)$	c
W	$\sum_{i=1}^{X} Ber(q_i(U_i))$	$\frac{K(2e^{ct_1t_3} - e^{ct_2t_3})}{46e^c}$
$Y \mid U, W$	$H(U, W, R)$	$\frac{WR}{U}$

sequence contained in the second urn. We have $W = \sum_{i=1}^{X} B_i$, where fix X, $\{B_i\}_{i=1}^{X}$ are independent Bernoulli random variables. Condition on $\{U_i\}_{i=1}^{K}$, the probability of success q_i of random variable B_i satisfies the following relation

$$q_i(U_i) \begin{cases} \geq 2(t_1t_3)^{U_i} - (t_2t_3)^{U_i} & \text{if } T \geq f, \\ = t_3^{U_i} & \text{otherwise,} \end{cases} \tag{2.1}$$

where $t_1 = \frac{L-T}{L-f}$, $t_2 = \frac{L-2T+f}{L-f}$, and $t_3 = 1 - \frac{f}{L}$, respectively. The proof is found in Appendix A.1.

Finally, conditional on U and W, the number of target sequences in the detection machine Y follows a Hypergeometric distribution with parameters (U, W, R).

2.3.1 Analytical calculations

In this section, we present the analytical results of our statistical modeling. Our goal is to set the sampling parameters K and R so that we can guarantee

$$P(Y \geq Q) \geq p, \text{ for pre-specified } Q \text{ and } p. \tag{2.2}$$

Now we consider R_{low}, such that with pre-fixed quantities p_0, U and W

$$P(Y \geq Q \mid U, W, R \geq R_{low}) \geq p_0. \tag{2.3}$$

Note here $Y|U, W \sim H(U, W, R)$. We will find R_{low} as a function of U, W, p_0 from tail bounds on Hypergeometric distribution.

In this chapter, we use the concentration inequality in Lemma A.3.2 on Binomial distribution to control the tail bounds of Hypergeometric distribution. Specifically, we will use the following theorem:

Theorem 2.3.1. *Let $h \sim H(A, B, C)$ be a hypergeometric random variable, $B_a \sim Bin(C, \frac{B}{A})$ and*

$B_b \sim Bin(A-C, \frac{B}{A})$ be two binomial random variables. Then under conditions on A, B, C and x listed in Appendix A.2, the following inequalities are true

$$P(h \leq x) \leq P(B_a \leq x), \tag{2.4}$$

$$P(h \leq x) \leq P(B_b \geq B - x). \tag{2.5}$$

The proof is in Appendix A.2. Numerical results presented in Section 2.5 suggest that for large C (2.5) is a better bound, in the remaining cases we will use (2.4).

Usually, one would want to fix (A, B, C) and calculate the tail bounds with different x. In this case only Property A.2.1 is needed to ensure the validity of Theorem 2.3.1. The remaining properties proved in Appendix A.2 ensure the validity of Theorem 2.3.1 for the other cases needed in Algorithm 1 when (A, B, C) are changing. In the subsequent calculations, we assume the conditions for Theorem 2.3.1 are met. In particular, we will use large deviation bounds from Lemma A.3.2 on the two Binomial distributions: $Bin(R, \frac{W}{U})$ and $Bin(U-R, \frac{W}{U})$ to find R_{low} in (2.3).

Write $R_{low} = R_{low}(U, W, p_0)$. Note that U and W are typically unknown. Therefore, R_{low} itself is still a random quantity and we need to further find a upper bound for R_{low} depending on n and K, this is denoted by $\hat{R}_{low}$. With large probability, sampling $\hat{R}_{low}$ sequences in the third urn is enough to guarantee sampling no less than R_{low} samples.

It is fairly straightforward to see R_{low} increases with W and decreases with U. Now we fix Q and p_0, and write U_{up} and W_{low} as the probabilistic upper/lower bounds for U and W, respectively. From (2.4) and (2.5) we can find $\hat{R}_{low}$ directly from tail bounds on $Bin(R, \frac{W_{low}}{U_{up}})$ and $Bin(U_{up}-R, \frac{W_{low}}{U_{up}})$. In particular, the steps needed to determine $\hat{R}_{low}$ for a given K and n are summarized here:

1. Use Lemma A.3.2 on Binomial distributions $Bin(K, \frac{1}{46})$ and $Bin(46n-K, \frac{1}{46})$ to find lower bound X_{low} of X. Here X_{low} depends only on n, K and p_1 so that: $P(X \geq X_{low}) \geq p_1$.

2. Set $X := X_{low}$ from step 1. Note that W is the summation of X_{low} independent Bernoulli trials. Hence from Lemma A.3.2 we can find lower bound W_{low} of W depending only on n, K, L, f, T, c, p_1, p_2 so that:$P(W \geq W_{low} \mid X \geq X_{low}) \geq p_2$. Consequently $P(W \geq W_{low}) \geq p_1 p_2$.

3. Use inequality from Lemma A.3.1 to find U_{up} and U_{low} depending only on c, K, p_3 so that: $P(U \geq U_{low}) \geq p_3$ and $P(U \leq U_{up}) \geq p_3$.

4. Use Lemma A.3.2 on binomial distributions $Bin(R, \frac{W_{low}}{U_{up}})$ and $Bin(U_{up} - R, \frac{W_{low}}{U_{up}})$ to find $\hat{R}_{low}$ so that

$$\begin{aligned} P(\hat{R}_{low} \geq R_{low}) &\geq P(U \leq U_{up}, W \geq W_{low}) \\ &\geq P(U \leq U_{up}) + P(W \geq W_{low}) - 1 \\ &= p_3 + p_1 p_2 - 1. \end{aligned}$$

Note that we need to ensure the needed sample size R is not larger than the available number of short sequences U. To this end, both $\hat{R}_{low}$ and U_{low} are deterministic functions of given constants and we can add numerical constraint on $\hat{R}_{low}$ to force it smaller than U_{low}. A key observation from our numerical result is, as K gets larger, U_{up} and U_{low} will be more concentrated around the mean $cK + K$, while R_{low} will be much smaller than U_{low}. Therefore, we need to find a lower bound K_{min} on K to ensure $U_{low} \geq \hat{R}_{low}$.

Finally, given that we choose K and $\hat{R}_{low}$ as our sampling sizes at two stages, respectively. The following relations are true

$$\begin{aligned} P(Y \geq Q) &\geq P(Y \geq Q, R \geq R_{low}, U \geq R) \\ &\geq p_0 \cdot P(\hat{R}_{low} \geq R_{low}, U \geq \hat{R}_{low}) \\ &\geq p_0 \cdot \Big[P(\hat{R}_{low} \geq R_{low}) + P(U \geq \hat{R}_{low}) - 1 \Big] \\ &\geq p_0(2p_3 + p_1 p_2 - 2). \end{aligned} \tag{2.6}$$

It suffices to set the desired probability p equal to the right-hand-side of (2.6). The exact selection of $\{p_i\}_{i=0}^3$ can be found in Section 2.4. We will also show in Section 2.4.1 that the range of K is $[K_{min}, 45n]$. While not every K in this range is feasible, a straightforward monotone analysis shows that as long as K is larger than a certain threshold, the solution $\hat{R}_{low}$ always exists.

2.3.2 Optimal sampling strategy

In this section, we discuss how to use the formulas derived in Section 2.3.1 to find the optimal values of n and K for any given p and Q. Specifically, assume there is a user-specified cost function $f(n, K)$ over number of samples n and the sampling size from first urn. In this chapter we assume

$f(\cdot,\cdot)$ is an monotone increasing function of both n and K.

The proposed procedure is summarized here:

1. Solve for $\{p_i\}_{i=0}^3$ such that $p = p_0(2p_3 + p_1p_2 - 2)$.
2. For fixed n, we calculate K_{min}.
3. For any fixed n and K such that $K \geq K_{min}$, we calculate $\hat{R}_{low}$.
4. Return: $(n, K, \hat{R}_{low})$.

The implementation details are discussed in Section 2.4. In reality the amount of biological materials is limited, hence there is an upper bound on n and there are only finite number of $(n, K, \hat{R}_{low})$ to consider. We do not need to consider any $R > \hat{R}_{low}$ as that would lead to sub-optimal design. However, for fixed n, we do need to consider $K > K_{min}$, because larger K might lead to smaller $\hat{R}_{low}$ and a more efficient solution.

Assume we have a cost function $C(K, R)$ that increases with K and R. We only have finitely many $(n, K, \hat{R}_{low})$ to consider and a brute force search among all the possible triples will yield the optimal $(n, K, \hat{R}_{low})$ minimizing the cost function.

Due to technology limits, we may have certain constraints on sampling percentages: for example, we can only sample 80% in the first stage, and 50% from the second stage. We can still use a brute force search only considering the cases that do satisfy these extra constraints.

2.4 Sampling algorithm

In this section we discuss the implementation details of optimal sampling strategy in Section 2.3.

The following quantities should be specified/calculated beforehand:

1. Specify the values of L, f, T, p, Q, n, c according to the particular application.
2. Select $p_0 = \sqrt{p}$, $3p_3 - 2 = \sqrt{p}$ and $p_1 = p_2 := \sqrt{p_3}$ so that the right-hand-side of (2.6) is p.
3. Compute: $t_1 = \frac{L-T}{L-f}$, $t_2 = \frac{L-2T+f}{L-f}$, $t_3 = 1 - \frac{f}{L}$ and set $Q_1 = \frac{2e^{ct_1t_3} - e^{ct_2t_3}}{e^c}$, $v = Q_1 - Q_1^2$. Here Q_1 and v are the expected value and variance of Bernoulli $\text{Ber}(q_i(U_i))$ random variable.

2.4.1 Lower bound on K

We need to find the lower bound K_{min} of K such that with large probability we have at least Q target sequences in the third urn. Equivalently, we want $R \geq Q$. To this end, we assume the

cutting process in urn 2 does not break any target sequences and we take everything out from urn 3. Therefore, we only need to make sure X is larger than Q with high probability. In Section 2.5, we solved both (2.4) and (2.5) to get different lower bounds for K, similarly with different lower bounds on K we will have different lower bounds for downstream quantities like X, U etc.

2.4.2 Lower bound on R

Algorithm 1 can be used to calculate $\hat{R}_{low}$ with pre-fixed n and K. Please note, that we use tail bounds of binomial distribution to approximate that of Hypergeometric distribution in step 1, 2 and step 4. Here steps 1 and 2 only require property 1 and 2 in Appendix A.2. Step 4 additionally needs property 3 and 4, because we need the relations in (2.4) and (2.5) to be true with both $W \geq W_{low}$ and $U \leq U_{up}$. For each fixed n, the range of K is relatively small, thus for each input n we can simply try all the possible K and calculate the corresponding smallest R (denoted by R_{low}) that achieves our goal. To make our algorithm more efficient, we can first find the smallest K that can give us a lower tail that is larger than Q (any smaller K will not be feasible, see our supporting code for details), call this K_{min}. For each K from K_{min} to $45n$, we use Algorithm 1 to find R_{low}.

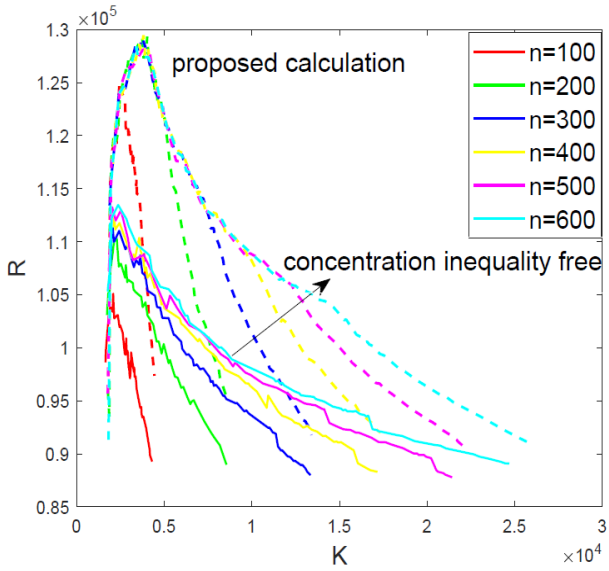
2.5 Numerical results

For our numerical results, the calculations were based on biologically reasonable parameters: $L = 250000000$, $f = 50000$, $T = 75000$, $c = 60$, $p = 0.95$, $Q = 20$.

In Figure 2.2, we plot our original calculation results from Algorithm 1 together with the results without using any concentration inequalities (we get the tail points by the inverse of cumulative distribution functions, which is applicable for relatively small n); both of them have the similar patterns. From original calculation results we can find two "kinks" for each fixed n. This is because when K is small, we will need to sample almost everything from the second stage, which will force us to choose the correspond $Bin(U_{up} - R, \frac{W_{low}}{U_{up}})$ for Y as the binomial bounds. Then as K gets larger but not big enough, we will use $Bin(R, \frac{W_{low}}{U_{up}})$ for both stages. Finally K will get close to $45n$ which again forces to use $Bin(U_{up} - R, \frac{W_{low}}{U_{up}})$ at the first sampling stage. The performance of our algorithm is slightly more conservative than the concentration inequality free approach in the sense that we ask for more samples. However, each lower bound of our algorithm can be solved efficiently using numerical method, while using inverse cdf function is generally slow for large n.

In Figure 2.3 we plot the simulation results together with population expectation results. Here the simulation means of each n and fixed K we create large amount of X, W and U. Then for

Figure 2.2: Approximation results of K vs $\hat{R}_{low}$



We use Algorithm 1 for n ranges from $n = 100$ to $n = 600$, different colors correspond to different n. Curves at the bottom correspond to concentration inequality free results, while dotted curves at the top correspond to results calculated from our algorithm.

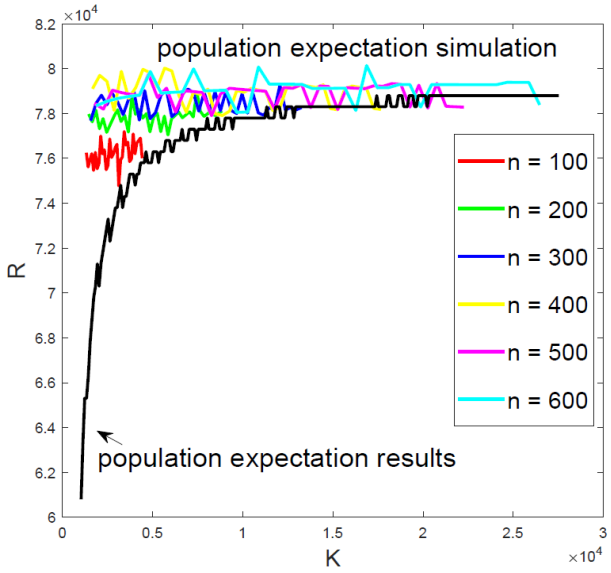


Figure 2.3: Simulation results and population expectation results

1: Apply Lemma A.3.2 to $B_a \sim Bin(K, \frac{1}{46})$ and $B_b \sim Bin(46n - K, \frac{1}{46})$. Solve the system

$$-log(c(1-p_1)) = Kh(\frac{t}{K} + 1 - \frac{1}{46}, 1 - \frac{1}{46}),$$
$$c = \max\{2, \sqrt{4\pi Kh(\frac{t}{K} + 1 - \frac{1}{46}, 1 - \frac{1}{46})}\},$$

and set $X_{low_1} = \frac{K}{46} - t$. Similarly we can solve for X_{low_2}. Set

$$X_{low} := \max(X_{low_1}, X_{low_2}).$$

2: Fix X to be X_{low}. Solve the following system

$$-log(c(1-p_2)) = X_{low}h(\frac{t}{X_{low}} + 1 - Q_1, 1 - Q_1),$$
$$c = \max\{2, \sqrt{4\pi X_{low}h(\frac{t}{X_{low}} + 1 - Q_1, 1 - Q_1)}\},$$

and set $W := Q_1 * X_{low} - t$.

3: Calculate the p_3 lower bound U_{low} and upper bound U_{up} for U from Lemma A.3.1.

4: Apply Lemma A.3.2 to $B_c \sim Bin(R, \frac{W_{low}}{U_{up}})$ and $B_d \sim Bin(U_{up} - R, \frac{W_{low}}{U_{up}})$, solve for R_{low_1}

$$r\frac{W_{low}}{U_{up}} - t = Q,$$
$$-log(c(1-p_0)) = rh(\frac{t}{r} + 1 - \frac{W_{low}}{U_{up}}, 1 - \frac{W_{low}}{U_{up}}),$$
$$c = \max\{2, \sqrt{4\pi rh(\frac{t}{r} + 1 - \frac{W_{low}}{U_{up}}, 1 - \frac{W_{low}}{U_{up}})}\},$$

set $R_{low_1} := r$. Similarly we can solve for R_{low_2}.

5: Set $\hat{R}_{low} = \min\{R_{low_1}, R_{low_2}\}$ and output $(n, K, \hat{R}_{low})$.

Algorithm 1: Compute $\hat{R}_{low}$ from fixed n and K

each simulation trial, we use a brute force search to find the smallest R that can gives us (2.2). Note this simulation is an "averaging" approach while our algorithm is more like a tolerance interval approach, thus they are not comparable and we put them into two separate figures. The population expectation results means we replace W and U directly by their expectations, and again brute force search for the smallest R. From Figure 2.3 we can see as K gets larger, these two results will be very close, hence for large K we can approximately use $\mathbb{E}[U]$ and $\mathbb{E}[W]$ to conduct the calculation.

Table 2.3 provides examples the minimization results based on a linear cost function, $C(K, R) = aK + bR$, under various constraints. In particular we use $a = 60$, $b = 1$ and various sampling percentage constraints on both sampling stages. Under all constraints, the algorithm tends to sample

Table 2.3: Minimization of cost function

Constraint 1	Constraint 2	n	K	R	$\frac{K}{46n}$	$\frac{R}{U_{low}}$
100%	50%	100	3652	111850	79.39%	49.96%
80%	20%	300	8716	106220	63.16%	19.91%
50%	100%	100	1812	91322	39.39%	82.03%
50%	50%	200	4136	126630	44.96%	49.96%
20%	80%	500	2572	124370	11.18%	78.8%

as many as possible in the second stage.

We have also applied our algorithm to other choices of Q. The lessons learned are similar to what we have shown here. In the supporting materials we provide Matlab code that can be used to calculate optimal sampling strategy with different parameters.

2.6 Conclusion

In this chapter, we have developed an optimization approach for estimating the amount of material needed for genomic mapping based on a simple, yet realistic, model of the process that uses a novel result regarding the tail bounds of the Hypergeometric distribution. Our approach is both computationally and analytically tractable and we show that *if* a genomics mapping technology can sample most of the chromosomal fragments within a sample, comparatively little biological material is needed to detect a variant at high confidence.

To bound the tails of hypergeomtric distribution, we borrowed the tail bounds from Binomial distribution and derived the conditions needed for such application. A direct tail bound on Hypergeometric distribution is developed in Appendix A.4.

CHAPTER 3

Semi-Parametric Model for Optical Mapping

3.1 Introduction

Early optical mapping technologies were limited in scope and scale of the genome that could be assessed. While microbial genomes could be readily assembled, more complex genomes were both technically and algorithmically challenging. A classic tool, Gentig, produced de novo assemblies without requiring an initial estimate of the genome-wide restriction map [Anantharaman et al., 1999], but was limited to small genomes. Other solutions included a heuristic assembler that uses pairwise Smith-Waterman alignment [Shi et al., 2016, Valouev et al., 2006b], subdividing the assembly problem in many smaller problems and using a low-level assembly engine [Mullikin and Ning, 2003]. Other approaches include methods originally developed from plasmid mapping Huddleston and Eichler [2016], Pendleton et al. [2015] and references therein.

Optical mapping technologies have since matured. Systems, such as the BioNano Saphyr, can characterize the order and organization of large genomes, such as the human genome [Udall and Dawe, 2018, Howe and Wood, 2015]. Today these approaches begin with an aliquot of cells isolated from the tissue of interest. The technology then performs an "on-chip" digestion of these cells, followed by extraction of the nucleic acids from these cells. These long DNA molecules (50-250 million base pairs [bp] of DNA per chromosome in humans) are then elongated either on a slide or in a nanochannel and probed for specific short DNA sequences within the the long sequences using either optical probes or restriction enzyme digest (nicking/cutting) based methods. These short sequences are usually short and found every 1000-100,000 bp. As the long sequence moves through the nanochannel each possible short sequence is evaluated beginning with the first bp. For a given sequence, this produces a list of lengths (bps) of distances between detected short sequences that is ordered by when they occurred in the initial long sequence. These lists of distances between the short sequences are compared to a reference generated from the human reference genome from that particular chromosomal region. Discrepancies between these two lists in the target sequence regions

potentially indicate a structural variant.

As the molecular technology of optical mapping has matured, the amount of data generated by these technologies has exponentially increased. Algorithms for aligning optical reads have focused on using fast computer science techniques to efficiently align the data. Dynamic programming and de Bruijn graphs, for example, are commonly used in these fast aligners [Valouev et al., 2006c, Luebeck et al., 2020, Fan et al., 2018]. The OMTools suite is an excellent example of these types of tools and provides a comprehensive set of fast algorithms and visualization tools [Leung et al., 2017a].

Optical genome mapping technologies are expected to be used for clinical diagnosis of SV associated with genetic disorders in the near future. In this scenario it is important to assess how well the optical mapping data support the presence of a variant and how well the data support the canonical reference sequence. For example, low confidence in the optical data supporting the reference sequence in a genomic region of interest may motivate further investigation of that region in an affected patient even if a variant is not called. Recognizing this issue, we developed a probabilistic semi-parametric approach for modeling the fit of optical mapping reads to a reference genome. Our approach allows us to assign each optical mapping read to its most likely location in the genome and rigorously assess the credibility of that assignment. Further, by developing an explicit statistical model for these procedures and implemented it as a open-source package, we make transparent both the model's assumptions and it's implementation. This work is a critical first step towards building a comprehensive analytical system for using optical mapping and similar data for characterizing structural variation in any genome.

3.2 Approach

Figure 3.1 summarizes our method. High molecular weight DNA is isolated and input into an optical mapping device. For a given molecule of DNA, the device yields an ordered series of lengths corresponding to the fragments of DNA in between the nicking or restriction endonuclease sites ("enzyme cutting sites"), which are short oligonucleotide sequences in the genome. If long enough and unique enough, the pattern of lengths in this ordered array should be characteristic of the region of the genome from which the DNA fragment derived. If the DNA sequence of the individual providing the DNA is known, then identifying the source genomic region is straightforward: nicking sites are predicted from the DNA sequence, lengths between enzyme cutting sites are calculated, and that inferred ordered list of lengths is compared to the experimentally observed. In practice, the

DNA sequence of the individual is not known. Instead, a reference genome, such as human Genome hg38, is used or generate the expected pattern of lengths. Again, the observed optical mapping fragment can then be matched to part of the reference array in order to identify the genomic region corresponding to the source of observed fragment. Several possible complications may affect how readily the observed fragment is mapped back to the reference. First, the DNA analyzed may harbor genetic variation causing it to differ from the reference genome. Second, diploid organisms harbor two copies of each chromosome, which may mean that neither regions is an exact match. Third, stochastic shearing of the DNA during the optical mapping may fragment the DNA. This may cause some fragments to be unobserved and will lead to erroneous lengths at the ends of the ordered arrays. Fourth, errors in the nicking or fragmenting process may lead to spurious lengths or missing sites. Finally, complex eukarotic genome are often rich in long arrays of repetitive elements that result in non-unique ordered arrays.

Here, our objective is to identify the "best" position within the genome for a set of observed fragments and provide robust estimate of the *quality* of the match. We build a statistical model to determine the quality of a position in the genome. In principle, this model could be applied to all fragments produced by an experiment, but the time needed to compute for any large dataset is long. We take advantage of the fact that most of the genome is a terrible fit for any one read. Using some conservative bounds and heuristics, and a rapid filtering algorithm we are able to accelerate the algorithm so that it can be applied to a human sized dataset.

3.3 Methods

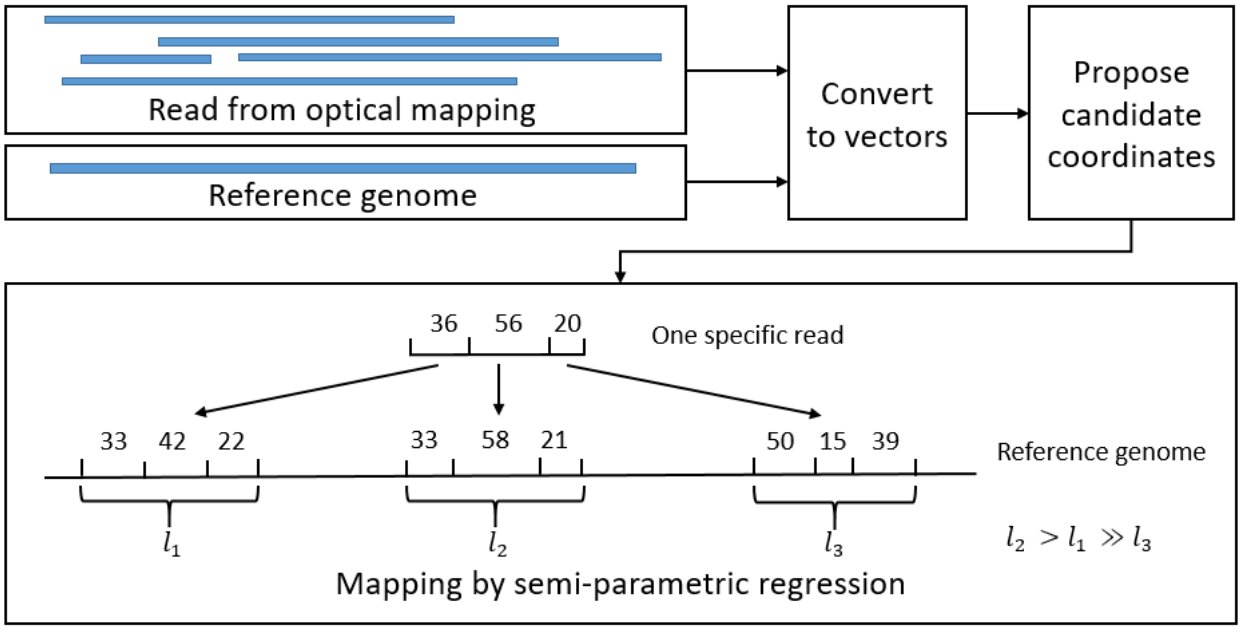
In this section we introduce a probabilistic approach in assessing the quality of alignments in optical mapping.

3.3.1 Model Setting

The optical mapping can be modeled through a mapping function from reference genome to observed reads. Specifically, we may write the reference genome as $[l_1, ..., l_N]$, here l_i represents the length (bp) between two adjacent enzyme cutting sites. Similarly we may write the observed reads as $[r_1, r_2, ..., r_m]$. We call $\{r_i\}_{i=1}^m$ and $\{l_i\}_{i=1}^N$ fragments and both of them are sets of positive integers.

During optical mapping, we cut the whole reference genome into several sub-sequences according to a Poison process with parameter λ. Each sub-sequence again can be represented as a vector of positive integers. Traditionally, the following mechanical errors are considered for optical mapping

Figure 3.1: Demonstration of optical mapping.



Demonstration of our method. The reference genome and reads are represented as vectors of positive integers. The filtering algorithm is used to propose the coordinates of candidates sequences. In the bottom part we have one specific read and three candidate sequences correspond to it. The $\{l_i\}_{i=1}^3$ are likelihoods calculated from fitted semi-parametric distribution, here we expect $l_2 > l_1 \gg l_3$.

1. Missing signals: enzyme sites get missed.

2. Extra signals: spurious extra enzyme sites appear in the genome.

3. Resolution error: fragments that are shorter than a threshold are not observable.

4. Sizing error: the original fragment and the mapped fragment have different lengths.

At this moment, we do not consider missing signals and extra signals. In Section 3.5, we will show the extension of our model to these two errors. Throughout this chapter, we assume

1. The optical mapping between fragments are completely independent.

2. Cutting happens exactly at the enzyme cutting sites.

3. Fragments generated by optical mapping that are shorter than a constant threshold T are not observable, this is the resolution error.

In this chapter, we call the sequence that mapped into the given observed read as original sequence, and call a sequence that might be the original sequence as a candidate sequence. The last assumption

means that the dimension of original sequence of $[r_1, ..., r_m]$ might be larger than m, i.e. short mappings are censored. For a given observed sequence $[r_1, ..., r_m]$, the likelihood that it was generated from candidate sequence $[l_{i_1}, ..., l_{i_k}]$ can be modeled as a posterior probability

$$\mathbb{P}([l_{i_1}, ..., l_{i_k}] \,|\, [r_1, ..., r_m]) \propto \mathbb{P}([r_1, ..., r_m] \,|\, [l_{i_1}, ..., l_{i_k}]) \cdot \mathbb{P}([l_{i_1}, ..., l_{i_k}]), \tag{3.1}$$

where $\{i_j\}_{j=1}^k \in \mathbb{N}^+$ are indexes in $[N]$, $m \leq k$. For the prior part we have

$$\begin{aligned}
\mathbb{P}([l_{i_1}, ..., l_{i_k}]) &= \sum_{q=2}^{\infty} \frac{\lambda^q e^{-\lambda}}{q!} \cdot q(q-1)c^{q-2} * L^{-2} \\
&= \left(\frac{\lambda}{L}\right)^2 \cdot \frac{e^{\lambda c}}{e^{\lambda}} \sum_{j=0}^{\infty} \frac{(\lambda c)^j e^{-\lambda c}}{j!} \\
&= \left(\frac{\lambda}{L}\right)^2 \cdot \frac{e^{\lambda c}}{e^{\lambda}},
\end{aligned} \tag{3.2}$$

where $q \sim Poi(\lambda)$ is the number of total cuttings in the reference genome (conditional on q, the cutting positions are uniformly distributed in $[0, L]$.), and $c = \frac{L-e+b}{L}$ ($L = \sum_{i=1}^N l_i$ is the physical length of reference genome, b and e are the physical starting and ending positions of $[l_{i_1}, ..., l_{i_k}]$ in the reference genome).

For the likelihood part we have

$$\mathbb{P}([r_1, r_2, .., r_m] \,|\, [l_{i_1}, ..., l_{i_k}]) = \begin{cases} \Pi_{j=1}^k g_{l_{i_j}}(r_j), & k = m \\ 0, & m > k \\ \sum_{q_1,..,q_m} \left[\Pi_{j=1}^m g_{l_{q_j}}(r_j) \cdot \Pi_{j \in \mathcal{I} \setminus \{q_1,..,q_m\}} \mathbb{P}_{l_j}(x \leq T)\right], & m < k, \end{cases}$$

where $\mathcal{I} = \{i_1, ..., i_k\}$, and $g_l(\cdot)$ is the density function of some unknown distribution that take l as the parameter. Note that $g_l(\cdot)$ catches the sizing error of optical mapping. In next section we will use the semi-parametric approach to model the density function.

3.3.2 Mapping Function via Semi-Parametric Generalized Linear Model

In this section, we develop a semi-parametric model for density function $g_l(\cdot)$. Traditionally, the sizing error is decomposed into two parts [Leung et al., 2017b, Shelton et al., 2015, Valouev et al., 2006a, Nagarajan et al., 2008, Muggli et al., 2014]: scaling error and measurement error.

Mathematically speaking, assume fragment r was generated from fragment l, then

$$r = (1 + s_e)l + m_e, \tag{3.3}$$

where s_e is the scaling factor, m_e is the measurement error. Previous works usually assume s_e and m_e are constants throughout ALL fragments. There are at least two issues with this heuristic approach: (1) the fragment lengths r and l are positive integers, but adding s_e and m_e violates this assumption; (2) the parametric approach are inflexible, especially because it assumes s_e and m_e are constants. In this chapter, we developed a semi-parametric GLM (SPGLM) model for the density function, which has higher flexibility and can approximate the true probabilistic nature of optical mapping.

Start from the uni-variate exponential density function

$$f(y|\theta) = exp\{\phi(y) + \theta y - A(\theta)\}. \tag{3.4}$$

In parametric exponential family, $\phi(y)$ is a parametric function of y, $A(\theta)$ is the normalizing constant and θ is called the natural parameter. In SPGLM, $\phi(\cdot)$ is modeled as a non-parametric function, and θ is a linear function of the co-variates X.

Use the same definitions for r and l as above. In stead of modeling the relationship between them as in (3.3), we consider the following standardized discrepancy between r and l

$$y = \frac{\sqrt{r} - \sqrt{l}}{\sqrt{l}}. \tag{3.5}$$

The standardization in (3.5) allows us to model the density function in a wider domain. Specifically, we model y as a random variable generated from the semi-parametric exponential family in (3.4), where $\phi(\cdot)$ is a non-parametric function on the support of y and $\theta = \beta_0 + \beta_1 \cdot x$, here $x = \sqrt{l}$. Our goal is to estimate $\phi(\cdot)$ and β.

Given a training set $\{(x_i, y_i)\}_{i=1}^n$, the maximum likelihood estimators of $\phi(\cdot)$ and β can be found via optimization methods. However, two issues arise: (1) we do not have the ground truth training set; (2) we need constraints on $\phi(\cdot)$ to make the optimization problem solvable.

Constructing Training Set In optical mapping, we only have access to the reference genome and thousands of observed reads. The exact position of original sequence for each observed read is unknown to us and technically impossible to locate with 100% confidence. Consequently, we do not have a "ground truth" training set at hand. In this dissertation, a heuristic minimization algorithm is used to construct the "heuristic training set".

The average relative error between a observed sequence $[r_1, .., r_m]$ and a candidate sequence $[l_{i_1}, ..., l_{i_m}]$ is calculated as

$$E = \frac{1}{m} \sum_{j=1}^{m} \frac{\sqrt{r_j} - \sqrt{l_{i_j}}}{\sqrt{l_{i_j}}}. \tag{3.6}$$

Our minimization algorithm takes an observed sequence (i.e. $[r_1, .., r_m]$) as input, and output the candidate sequence in the reference genome that delivers the smallest average relative error calculated from (3.6). Note that we only consider candidate sequence with same dimension as the input, i.e. $k = m$. To get the training set, we apply the heuristic minimization algorithm to 1000 reads extracted from real experiments. The sequences that have minimal smallest average relative errors are used to construct the training set. Specifically, we use the fragments in these reads and their corresponding fragments in the reference genome found by the heuristic minimization algorithm as the "ground truth". Finally we have a sample with 3000 standardized (see (3.5)) training samples (x_i, y_i).

Constraints on $\phi(\cdot)$ Given a training set $\{(x_i, y_i)\}_{i=1}^n$. We write $\{y_{(i)}\}_{i=1}^n$ as the order statistics of $\{y_i\}_{i=1}^n$ in ascending order. The range $[y_{(1)}, y_{(K)}]$ is used as the support of our semi-parametric distribution, here $K \leq n$ since we may have duplicates.

In this dissertation, function $\phi(\cdot)$ is modeled as a concave piece-wise linear function that only changes its slope at $\{y_{(i)}\}_{i=1}^{(K)}$. Write $\phi_k = \phi(y_{(k)})$, then for $\forall y \in \mathbb{R}$ we have

$$\phi(y) = \begin{cases} \left(1 - \frac{y - y_{(k)}}{y_{(k+1)} - y_{(k)}}\right) \phi_k + \frac{y - y_{(k)}}{y_{(k+1)} - y_{(k)}} \phi_{k+1}, & \text{if } y \in [y_{(k)}, y_{(k+1)}] \\ -\infty, & \text{o.w.} \end{cases}$$

Let $\Delta_k = y_{(k+1)} - y_{(k)}$. To assure the concavity of $\phi(\cdot)$, we need for $\forall 2 \leq k \leq K - 1$

$$-\frac{1}{\Delta_{k-1}} \phi_{k-1} + \left(\frac{1}{\Delta_{k-1}} + \frac{1}{\Delta_k}\right) \phi_k - \frac{1}{\Delta_k} \phi_{k+1} \geq 0. \tag{3.7}$$

Write $\alpha = B \cdot \phi$, where

$$B = \begin{bmatrix} 1 & -\frac{1}{\Delta_1} & 0 & \cdots & 0 \\ -\frac{1}{\Delta_1} & \cdots & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & -\frac{1}{\Delta_{i-1}} & \frac{1}{\Delta_{i-1}} + \frac{1}{\Delta_i} & -\frac{1}{\Delta_i} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & \frac{1}{\Delta_{K-1}} & 1 \end{bmatrix}.$$

The concavity constraint can be rewritten: $\alpha_k \geq 0$ for $k = 2, .., K-1$.

To make our estimation identifiable, we also add the following identifiability constraints

$$\int_{y_{(1)}}^{y_{(K)}} e^{\phi(y)} dy = 1, \quad \int_{y_{(1)}}^{y_{(K)}} y e^{\phi(y)} dy = 1. \tag{3.8}$$

See [Zhang, 2017, Chapter 4] for a detailed discussion on why these constraints are needed and why they work. The numerical set up of these constraints is discussed in Section B.1.

3.3.3 Optimization with Projected Gradient Descent Algorithm

To find the MLE of $\phi(\cdot)$ and β, we want to maximize the following objective function

$$L_n(\phi, \beta) = \Pi_{i=1}^N exp\{\theta_i y_i + \phi(y_i) - A(\theta_i)\}, \tag{3.9}$$

where

$$A(\theta_i) = \log \int_{y_{(1)}}^{y_{(K)}} e^{\theta_i y + \phi(y)} dy.$$

Note that $L_n(\phi, \beta)$ is the likelihood function of SPGLM. We solve the optimization problem in a projected gradient descent fashion, i.e. we first do the steepest descend on ϕ, and then project it back to the feasible region, then do the same procedure for β, we iterate these two procedures until the results converge. Handling identifiability modification is tricky, we leave the technical details in Appendix B.1.

Write $l_n(\cdot)$ as the log-likelihood function. In summary, the optimization procedure contains the following main steps

1. Initialize β and ϕ.

2. Fixing $\hat{\phi}$, update $\hat{\beta}$ by doing gradient descent with respect to

$$l_n(\beta) = \sum_{i=1}^{n} \{x_i^T \beta y_i + \hat{\phi}(y_i) - \log \sum_{k=1}^{K} exp(\hat{\phi}_k + x_i^T \beta y_k)\}. \tag{3.10}$$

3. Fixing $\hat{\beta}$, update $\hat{\phi}$ by doing gradient descent with respect to

$$l_n(\phi) = \sum_{i=1}^{n} \{x_i^T \hat{\beta} y_i + \phi(y_i) - \log \sum_{k=1}^{K} exp(\phi_k + x_i^T \hat{\beta} y_k)\}. \tag{3.11}$$

4. Update $\hat{\phi}$ by projecting it back into the feasible region.

5. Iterate over Step B to Step D until convergence.

6. Modify $\hat{\phi}$ by the following equation

$$\hat{\phi} := \hat{\phi} + \theta^* y - \log \int_0^\infty e^{\hat{\phi} + \theta^* y} dy. \tag{3.12}$$

   Here θ^* satisfies the following condition

$$\frac{\int_{y_{(1)}}^{y_{(K)}} y e^{\hat{\phi} + \theta^* y} dy}{\int_{y_{(1)}}^{y_{(K)}} e^{\hat{\phi} + \theta^* y} dy} = 1.$$

   We can use binary search to numerically solve for θ^*. It is fairly straightforward to check this modification can actually meet the identifiability constraints in (3.8).

The closed formulas needed for optimization are presented in Appendix B.1. The implementation details can be found in supplementary codes.

3.3.4 A Boolean-Matrix-based Filtering Algorithm

Generally speaking, any subsequence $[l_{i_1}, ..., l_{i_k}]$ of the reference genome where $k \geq m$ could be a valid candidate sequence of $[r_1, ..., r_m]$. However, calculating the posterior likelihoods with respect to all these potential candidates is often prohibitive with the limited computational resources. In this section, we proposed a fast filtering algorithm that can eliminate out most of the candidates that are unlikely to generate the given observed read.

The filtering algorithm starts with a boolean matrix (matrix with only 1's and 0's as its entries) $\mathbf{B} \in \mathbb{R}^{m \times N}$ such that

$$\mathbf{B}_{ij} = \begin{cases} 1, & \text{if } \frac{\sqrt{r_i}-\sqrt{l_j}}{\sqrt{l_j}} \in [y_{(1)}, y_{(K)}]. \\ 0, & \text{otherwise.} \end{cases}$$

In words, $\mathbf{B}_{ij} = 1$ if and only if it is feasible that l_j to map into r_i. If no drops are allowed, a sequence $[l_{i_1}, .., l_{i_m}]$ is a feasible potential original sequence if and only if $\Pi_{j=1}^{m} \mathbf{B}_{j,i_j} = 1$.

Under the semi-parametric distribution, a fragment l_{i_j} can be dropped if and only if

$$\frac{\sqrt{T} - \sqrt{l_{i_j}}}{\sqrt{l_{i_j}}} \in [y_{(1)}, y_{(K)}]. \tag{3.13}$$

Therefore we can pre-calculate a vector $\mathbf{v} \in \mathbb{R}^N$ such that

$$v_j = \begin{cases} 1, & \text{if (3.13) is satisfied.} \\ 0, & \text{otherwise.} \end{cases} \tag{3.14}$$

Vector $\mathbf{v}$ stores the set of fragment indexes that can be dropped.

Definition 3.3.1. *A sequence $[l_{i_1}, ..., l_{i_k}]$ is a feasible candidate with respect to read $[r_1, ..., r_m]$ if and only if: there exists a subset $\{s_1, s_2, ..., s_m\}$ of $[i_1, .., i_k]$ such that*

$$\Pi_{j=1}^{m} \mathbf{B}_{j,s_j} \cdot \Pi_{i \in [i_1,..,i_k] \setminus \{s_1,...,i_m\}} v_i = 1.$$

We call the number of drops allowed as the dropping threshold. The filtering algorithm takes one read and a dropping threshold d as the input, and outputs all the feasible candidates with dimension $k = m + d$. The detailed steps of the filtering algorithm are listed in Algorithm 2. Please note that in Step 3, j is not a positive integer, instead it is a tuple that contains one possible combination of dropping positions. For example, when $k = m + 1$, the dropping position could happen at any position between 2 and m (we do not consider the first dimension nor the last dimension, since it essentially degenerates to the case $k = m$), so j is a list that contains just one positive integer; when $d = 2$, each j contains two positive numbers.

input: The reference genome L, the observed read $[r_1, ..., r_m]$, the dropping threshold d, the corresponding boolean matrix $\mathbf{B}$, and the dropping index vector $\mathbf{v}$.
output: The matrix $\mathbf{C}$ that stores the starting and ending positions for all feasible candidates.
1. Calculate $\mathcal{J}$ as the set that contains all the combinations of possible positions of drops, so the cardinality of $\mathcal{J}$ is $\binom{m+d-2}{d}$.
2. **for** $j \in \mathcal{J}$ **do**
  for $i = 1$ *to* $N - m - d + 1$ **do**
    $[j_1, ..., j_m] := \{i, i+1, ..., i+m+d-1\} \setminus (i+j)$
    if $\Pi_{s=1}^{m} \mathbf{B}_{s,j_s} \Pi_{t \in j} v_t == 1$ **then**
      $\mathbf{C} = [\mathbf{C}; i, i+m+d-1]$
    end
  end
end

Algorithm 2: Boolean Matrix Based Filtering

Pre-filtering of reference genome One drawback of Algorithm 2 is, the dropping threshold d is prefixed. This limits our searching space in the whole reference genome. One solution to this issue is to try different d and merge the outputs. While it is relatively cheap to run Algorithm 2 with small d, the cardinality of $\mathcal{J}$ in Algorithm 2 makes large d prohibitive. In this section, we discuss the pre-filtering of reference genome, which can significantly increase the searching space of Algorithm 2.

To pre-filter the reference genome, we remove all the fragments in the reference genome with lengths smaller than a threshold $c(T)$. For any specific original sequence, we can divide its fragments into three categories: (1) fragments that map into the observable reads, hence the number of fragments in this category equals to the dimension of observed read, (2) fragments that are shorter than $c(T)$ and map into something un-observable, (3) fragments that are longer than $c(T)$ and map into something un-observable.

In Algorithm 2, we allow exact d drops happen in the candidate sequences, hence d is the total number of fragments in the second and third category above. However, when $c(T)$ is small enough, the probability that something below $c(T)$ maps into a fragment shorter than T is almost 1 (we can calculate the value of $c(T)$ based on the semi-parametric distribution), hence as an approximation we may just remove all the fragments with lengths shorter than $c(T)$. By doing so the number d is used to bound the number of sequences in category (3) above, and hence we implicitly allow more

Table 3.1: Statistics for different cutters.

	# of Fragments	Mean	Median	Max	Min
Cutter_1	693849	436	2114	18164240	6
Cutter_2	192920	15711	8869	19848848	6
Cutter_3	186644	16240	9948	18171066	7

drops and add flexibility to the algorithm. Define $c(T)_p$ as a constant such that

$$\mathbb{P}\left[x \leq \frac{\sqrt{c(T)_p} - \sqrt{T}}{\sqrt{c(T)_p}}\right] \leq 1 - p. \tag{3.15}$$

Therefore $c(T)_p$ is monotonically increasing in p. Write L_p as the vector after removing all the fragments $\{l_i\}_{i\in[N],l_i<c(T)}$ in reference genome L, we call L_p the "filtered genome". Applying Algorithm 2 on L_p with different p yields faster calculation since we only need to build boolean matrix B on a shorter vector.

To sum up. In order to avoid narrowing down the searching space of Algorithm 2, we recommend applying Algorithm 2 on L_p with different choices of (p, d) and merge all the candidate sequences output by different parameters.

3.4 Numerical Results

In this section, we present our numerical results in two experiments with synthetic optical reads. In the first experiment, we test the performance of our algorithm on a human reference genome with different enzyme cutters. In the second experiment, we compare our algorithm with OMBlast on synthetic reads generated from E.Coli reference genome.

3.4.1 Generating Synthetic Reads

To test the performance of our method, three reference genomes were generated by applying three different cutters on the same human reference genome. In Table 3.1 we summarize the statistics over the lengths of fragments correspond to each cutter. Since the cutters are used on the same human genome, all of these genomes have same lengths and larger number of fragments means more short fragments.

For each cutter, we randomly sample 1000 original sequences from it. The physical length of each sampled sequence is determined based on the lengths of our true optical reads.

Now we have 1000 original sequences for each reference genome, the next step is to map each of the original sequence by three different mapping mechanisms to generate 3000 synthetic reads for

each cutter. To be specific, we model the relative error by (1) Negative binomial with $p = 0.5$, (2) Normal distribution with $\mu = 0$ and $\sigma = 0.1$, (3) semi-parametric distribution we fitted.

3.4.2 Details of Implementation

In this section, we discuss the details of our implementation of SPGLM and the filtering algorithm.

Phase 1: Filtering Algorithm In Phase 1, we set $T = 400$. This number is determined by the true nature of Bionano optical mapping device. Recall that we want to apply Algorithm 2 on L_p with different (p, d). In our experiments, we pick $d = 1, 2, 3$ and $p \in [0.11, 1]$ (this corresponds to $c(T)_p \in [40, 600]$).

Phase 2: Calculating Posterior In Phase 2, we set the mean of the poisson cutting procedure $\lambda = \frac{\sum_{i=1}^{N} l_i}{250000}$, here 250000 is the average physical length of optical reads and is determined by the Bionano optical mapping device. Note that λ only affects the prior.

To speed up the calculation of likelihood function, we use a lower bound of the posterior probability to approximate the full posterior. Specifically, we observed the fact that in real experiments, the true alignment in

$$\sum_{q_1,..,q_m} \left[\Pi_{j=1}^{m} g_{l_{q_j}}(r_j) \cdot \Pi_{j \in \mathcal{I} \setminus \{q_1,..,q_m\}} \mathbb{P}_{l_j}(x \leq T) \right]$$

usually dominates all the other possibilities. To this end, for each candidate sequence $[l_{i_1}, ..., l_{i_k}]$, we use a dynamic programming approach to find the set $\{\hat{q}_i\}_{i=1}^{m}$ such that

$$\{\hat{q}_i\}_{i=1}^{m} = \underset{\{q_i\}_{i=1}^{m} \in \mathcal{I} = \{i\}_{i=1}^{k}}{\arg\min} \sum_{i=1}^{m} \left| \frac{\sqrt{r_i} - \sqrt{l_{q_i}}}{\sqrt{l_{q_i}}} \right|.$$

The term $\Pi_{j=1}^{m} g_{l_{\hat{q}_j}}(r_j) \cdot \Pi_{j \in \mathcal{I} \setminus \{\hat{q}_i\}_{i=1}^{m}} \mathbb{P}_{l_j}(x \leq T)$ is then used to approximate the likelihood part of the posterior. We can summarize our implementation by the following steps

1. Pick a set $\{p_{t_j}\}_{j=1}^{s}$, and $d = 1, 2, 3$.
2. For every $p_{t_j} \in \{p_{t_j}\}_{j=1}^{s}$, get $L_{p_{t_j}}$ from L based on $c(T)_{p_{t_j}}$.
3. Calculate vector $\mathbf{v}$ from (3.14) based on $L_{p_{t_j}}$.
4. Generate binary matrix $\mathbf{B}$ based on $L_{p_{t_j}}$ and the specific read $\mathbf{r} = [r_1, ..., r_m]$.

5. Implement Algorithm 2 with dropping threshold d, filtered reference genome $L_{p_{t_j}}$, optical read $\mathbf{r}$, and vector $\mathbf{v}$. Assume we get candidate set $\mathbf{C}_{p_{t_j},d}$.

6. Merge the candidate sets $\mathcal{C} = \cup_{j=1}^{s} \mathbf{C}_{p_{t_j},d}$.

7. Calculate the posterior of every candidate in $\mathcal{C}$.

Results on Synthetic Reads from Human Reference Genome After Phase 2, we can rank the candidate sequences according to their corresponding posterior likelihoods. Each candidate sequence can be uniquely identified by its starting and ending positions in the reference genome (not the filtered reference genome). For any two sequences with starting and ending positions (s_1, e_1) and (s_2, e_2), we use $|s_1 - s_2| + |e_1 - e_2|$ to measure the physical distance between them. Since we are doing numerical analysis on synthetic data, the exact starting and ending positions of each original sequence are known to us. Therefore, for each read, we can calculate the physical distances between its corresponding original sequence and the candidate sequences found by the algorithm. We call the candidate sequence that has the smallest physical distance with the original sequence as the "closest match".

The numerical results is summarized in Table 3.2. Due to the space limitation, we use short names for each column. The corresponding descriptions of columns are under the table. Two desired properties regarding the closest match are

1. The distance between closest match and the true original sequence is small. In Table 3.2 we use "D" to denote this distance, note D is non-negative. A small D means our algorithm finds a candidate close to the truth.

2. The posterior likelihood of closest match is large. Note that even if closest match is the original sequence, to actually pick it out from large amount of candidates, we are more likely to pick it if it is inside some confidence region. In Table 3.2 we use "Post" to denote its posterior likelihood. Alternatively, we want the closest match to have high rank among candidate sequences. This is represented by the third column in Table 3.2, which demonstrates the portion of closest matches that rank top 10 among the candidates.

Algorithm 2 can deliver better results with respect to the second and third cutters in the sense that, in larger percentage of cases we can successfully find the true original sequence. While our

Table 3.2: Results on human reference genome

Generating distribution	# Reads	D = 0	D ≤ 2	Post >0.9	Rank top 10
Normal	995	59.20%	91.06%	62.81%	90.25%
NB	997	61.08%	91.88%	72.02%	94.28%
Semi-Para	985	53.71%	82.03%	36.04%	68.02 %
Normal	998	91.28%	98.70%	90.38%	98.8 %
NB	998	92.59%	99.30%	94.79%	99.8%
Semi-Para	982	89.51%	96.64%	72.40%	96.23%
Normal	996	93.37%	99.60%	92.37%	98.9%
NB	998	93.29%	99.60%	95.09%	98.5%
Semi-Para	985	88.43%	96.95%	73.40%	96.04%

Each block corresponds to the same reference genome cut by a specific enzyme cutter. For each genome, three different random mechanisms are used to generate the observed reads. In the second column, we record the number of outputs for each category (due to running time limit, we did not get the results for some reads). The third and fourth columns record the percentage of cases where the closest match is close to the true original sequences. The fifth column is the percentage of cases where closest match has a posterior likelihood larger than 0.9. The last column is the percentage of cases where closest match has a posterior likelihood rank top 10 among all candidate sequences.

algorithm does not take into account the structural variant at this moment, we can potentially use it for data with structural variant as well.

Comparison with OMBlast Mapping Tool In this section, we compare the performance of our algorithm with another state-of-the-art mapping tool called OMBlast [Leung et al., 2017b] on the synthetic datasets generated from Escherichia coli reference genome.

Specifically, we generated synthetic reads from two different types of data generation mechanisms. The first type is from OMBlast's data generator, where sizing error is decomposed into scaling error and measurement error. Note that OMBlast is capable of handling missing and extra signals, which is not supported by our algorithm at this moment. Therefore in the simulation we disabled the generation of missing and extra signals. The second type is the same as the previous section, where sizing error is modeled through relative error in (3.5). Similar as before, we generate the relative error based on three different generating distributions. We also added backward reads to make fair comparisons.

The results are summarized in Table 3.3. To decide the direction of mapping from our algorithm, for each specific read $[r_1, .., r_m]$ we run our algorithm on both $[r_1, .., r_m]$ and $[r_m, .., r_1]$. The normalized constants based on two runs are then used to decide the direction of the mapping.

Table 3.3: Comparison with OMBlast

Generating mechanism	# Reads	Semi Strong	Semi Weak	Semi All	OMB All
OMB	2995	95.53%	0.03%	95.56%	97.83%
Semi-Para	3000	84.53%	0.3%	84.83%	9.1%
Normal	3000	92.4%	0.03%	92.43%	34.03%
NB	3000	94.47%	0.07%	94.53%	90.33%

The first column records the data generation mechanisms, where first raw comes from data generator of OMBlast, second to fourth rows come from relative error with different generating distributions. The second column records the number of synthetic reads under different mechanisms. The third column records the cases where our algorithm find the true original sequences among the top 10 candidate sequences, while the fourth column records the cases where our algorithm finds the true original sequence but the direction is incorrect. The fifth column is the sum of the third and fourth columns. The last column records the percentages of cases where OMBlast successfully find the original sequences.

Note that OMBlast occasionally output the partial alignments. In Table 3.3, we deemed that OMBlast successfully finds the original sequence if and only if: (1) the partial alignment has overlapping with true original sequence, (2) at least half of the read is correctly mapped back to the original sequence.

From Table 3.3 we can see that our algorithm performs consistently well across different generating mechanisms, while OMBlast performs very poorly with medium to large sizing errors. The biggest advantage of our approach over OMBlast is, once the semi-parametric distribution is fitted from data, we do not need to specify any hyper-parameters. While one needs to specify the parameters for OMBlast that controls its tolerance to sizing error.

3.5 Extension to Other Mapping Tools

In this section, we extend the likelihood function in (3.1) to account for missing/extra signals. From this extension, we could potentially stack our semi-parametric likelihood model and other filtering algorithms (OMBlast Leung et al. [2017b], TWINS Muggli et al. [2014], SOMA Nagarajan et al. [2008] etc.) together.

We start from the following assumptions

1. Each enzyme site has independent chance of missing, the probability of missing is p_m.

2. The number of extra signals in a sequence with length $\sum_{j=1}^{k} l_{i_j}$ follows a Poisson distribution with mean $\lambda_e \sum_{j=1}^{k} l_{i_j}$, here λ_e is the extra signal rate.

3. The number of missing signals and extra signals are known to us. The sequence $[l_{i_1}, l_{i_2}, ..., l_{i_k}]$ is observed AFTER we remove the missing sites and add the extra sites.

Now for observed read $[r_1, ..., r_m]$ and candidate sequence $[l_{i_1}, l_{i_2}, ..., l_{i_k}]$, write n_m and n_e as the number of missing/extra signals, and $t_l = \sum_{j=1}^{k} l_{i_j}$. We then have

$$\mathbb{P}\left([l_{i_1}, ..., l_{i_k}]\right) \propto \left(\frac{\lambda}{L}\right)\left(\frac{e^{\lambda c}}{e^{\lambda}}\right)\left(\frac{p_m}{1-p_m}\right)^{n_m}\left(\frac{(\lambda_e t_l)^{n_e} e^{-\lambda_e t_l}}{n_e!}\right). \tag{3.16}$$

Here we add missing/extra signals by Bernoulli trials and Poisson distribution.

In reality, we only have access to L before removing the missing signals and adding extra signals, which is the opposite of Assumption C above. However, alignment tools like OMBlast is able to infer the missing and extra signals. Specifically, OMBlast outputs the sequence $[l_{i_1}, ..., l_{i_k}]$ together with the inferred positions of missing/extra signals. From these information, we can directly get the candidate sequence after add missing/extra signal errors, this is our Assumption C.

To sum up, with the modification in (3.16), we can use (3.1) to directly calculate the likelihoods of every candidate sequence output by OMBlast and other filtering algorithms.

CHAPTER 4

Subspace Clustering through Sub-Clusters

4.1 Introduction

In data analysis, researchers are often given data sets with large volume and high dimensionality. To reduce the computational complexity arising in these settings, researchers resort to dimension reduction techniques. To this end, traditional methods like PCA [Hotelling, 1933] use few principal components to represent the original data set; factor analysis [Cattell, 1952] seeks to get linear combinations of latent factors; subsequent works of PCA include kernel PCA [Schölkopf et al., 1998], generalized PCA [Vidal et al., 2005]; manifold learning [Belkin and Niyogi, 2003] assumes data points collected from a high dimensional ambient space lie around a low dimensional manifold, and muli-manifold learning [Liu et al., 2011] considers the setting of a mixture of manifolds. In this dissertation, we focus on one of the simplest manifold, a subspace, and consider the subspace clustering problem. Specifically, we approximate the original dataset as an union of subspaces. Representing the data as a union of subspaces allows for more computationally efficient downstream analysis on various problems such as motion segmentation [Elhamifar and Vidal, 2009], handwritten digits recognition [You et al., 2016a], and image compression [Hong et al., 2006].

4.1.1 Related Work

Many techniques have been developed for subspace clustering, see Vidal [2010] for a review. The mainstream methods usually include two phases: (1) calculating the affinity matrix; (2) applying spectral clustering [Ng et al., 2002] to the affinity matrix to compute a label for each data point. For phase (1), the property of self-representation is often used to calculate the affinity matrix: self-representation states that a point can be represented by a linear combination of other points in the same subspace. Specifically, Elhamifar and Vidal [2009] proposed the sparse subspace clustering (SSC) algorithm which solves the lasso minimization problem N times, where N is the total number of data points. Similarly, Rahmani and Atia [2017] proposed the direction search algorithm (DSC) which uses ℓ_1 minimization to find the "optimal direction" for each data point, these directions

are then used to cluster the data points. One of the main drawbacks of SSC and DSC is their computational complexity of $O(N^2)$ in both time and space, which limits its application to large datasets. To address this limitation, a variety of methods have been proposed to avoid solving complicated optimization problems in constructing the affinity matrix. Heckel and Bölcskei [2015] used inner products with thresholding (TSC) to calculate the affinity between each pair of points, Park et al. [2014] used a greedy algorithm to find for each point the linear space spanned by its neighbors, similarly Dyer et al. [2013] and You et al. [2016c] used orthogonal matching pursuit (OMP), You et al. [2016b] used elastic the net for subspace clustering (ENSC) and proposed an efficient solver by active set method. However, these methods require running spectral clustering on the full $N \times N$ affinity matrix. A Bayesian mixture model was proposed for subspace clustering in Thomas et al. [2014], but its parameter inference is not scalable to large dataset. Zhou et al. [2018] used a deep learning based method which does not have theoretical guarantee.

Recently, there have been two methods that increase the scalability of sparse subspace clustering. The SSSC algorithm and its varieties [Peng et al., 2016] clusters a random subset of the whole dataset and then uses this clustering to classify or label the out-of-sample data points. This method scales well when the random subset is small, however a great deal of information is discarded as only the information in the subset is used. In You et al. [2016a] a divide and conquer strategy is used for SSC—the data set is split into several small subsets on which SSC is run, and clustering results are merged. This method cannot reduce the computational complexity of the SSC by an order of magnitude so is limited in its ability to scale to large dataset.

4.1.2 Contribution

In this chapter, we propose a novel, efficient sampling based algorithm with provable guarantees that extends the ideas in previous scalable methods [Peng et al., 2016, You et al., 2016a]. A key observation driving our algorithm is the fact that only a small fraction of the original dataset is needed to recover the membership of each point, hence clustering a subset of the data should be adequate. In particular, for each point in the subset we find its nearest neighbors in the complete dataset and use these points to construct a sub-cluster, these sub-clusters contain information from the entire dataset and not just the random sample. The affinity matrix for the subset is then constructed from these sub-clusters. The idea is that we change the problem from "clustering of data points" to "clustering of sub-clusters", which integrates information across the dataset and should

deliver better clustering results.

We provide theoretical guarantees for our procedure in Section 4.3. The analysis reveals that under mild conditions, the subspaces can share arbitrarily many intersections as long as most of their principal angles are larger than a certain threshold. While our algorithm for finding neighboring points is similar to that of Heckel and Bölcskei [2015], the data generation model and assumptions underlying our theorems are different–we take into account the fact that after normalization the noisy terms will no longer follow a multivariate normal distribution. While our work is originally designed for linear subspace clustering problems. The idea of clustering through sub-clusters can be easily extended our to general clustering problems. See our discussion in Section 4.2.1.

We apply our algorithm to both synthetic and real world datasets. The experimental results demonstrate that our method is highly scalable and can deliver superior accuracy compared to other state-of-the-art methods.

4.1.3 Chapter Organization

The rest of this chapter is organized as follows: in Section 4.2, we describe the model setting and the implementation of our clustering procedure, in Section 4.3 we state theoretical guarantees for our procedure and explain in some details the geometric and distributional intuitions underlying our procedure. The detailed proofs can be found in Appendix C.1, in Section 4.4 we present experiments on four datasets and compare our method with state-of-the-art methods, a comprehensive report of the numerical results can be found in Appendix C.3.

4.1.4 Notation

Throughout this chapter, unless specified otherwise, we use capital bold letter to denote data matrix, and corresponding lower bold letter to denote the columns of it. In this chapter, we are given a dataset $\mathbf{Y}$ with N data points in $\mathbb{R}^D$. Write $\mathbf{y}_i$ as the i-th column of $\mathbf{Y}$, and $\mathbf{Y}_{-i}$ is the matrix $\mathbf{Y}$ with the i-th column removed. Similarly, we write $\mathbf{y}_{-j}$ as vector $\mathbf{y}$ with the j-th entry removed. The complement of event $\mathcal{E}$ is denoted by $\mathcal{E}^{\complement}$. We use subscript with parenthesis to represent the order statistics of entries in a vector, for example $\mathbf{a}_{(i)}$ is the i-th smallest entry in vector $\mathbf{a}$, while without ambiguity both $\mathbf{a}(i)$ and a_i refer to the i-th element of vector $\mathbf{a}$. The unit sphere in $\mathbb{R}^d$ is denoted by$\mathbb{S}^{d-1}$. We assume each data point of $\mathbf{Y}$ concentrates near exactly one of K linear subspaces denoted by $\{\mathcal{S}_k\}_{k=1}^K$. Here K is a known constant and $\mathcal{S}_k$ is the k-th linear subspace. The subspace clustering problem aims assigning to each point in $\mathbf{Y}$ membership to a subspace (cluster)

$\mathcal{S}_k$.

We write d_k as the dimension of subspace $\mathcal{S}_k$ and $\mathbf{U}_k \in \mathbb{R}^{D\times d_k}$ as its corresponding orthogonal base. The number of points belong to cluster $\mathcal{S}_k$ is N_k. We use $\mathbf{y_i}^{(k)} \in \mathbb{R}^D$ to represent a single point from the k-th cluster, the set $\{\mathbf{y_1}^{(k)}, ..., \mathbf{y_{N_k}}^{(k)}\}$ contains all points that belong to $\mathcal{S}_k$. Finally, we write $F_{m,n}$ as the F distribution with parameters (m, n), $Dir(\boldsymbol{\alpha})$ as the Dirichlet distribution with parameter vector $\boldsymbol{\alpha}$, and $\beta(a, b)$ as the Beta distribution with parameters (a, b).

4.2 Sampling Based Subspace Clustering

In this section, we introduce our sampling based algorithm for subspace clustering (SBSC). The detailed steps of this algorithm will be presented in Section 4.2.1. In Section 4.2.2 we discuss the issues regarding hyper-parameters. In Section 4.2.3 we provide comments on both the intuitions underlying the procedure and the advantages of our procedure. Throughout this section, we assume the columns of $\mathbf{Y}$ have unit ℓ norm.

4.2.1 The Algorithm for Sampling Based Subspace Clustering

Our main algorithm takes the raw data set $\mathbf{Y}$ and several parameters as input and outputs the clustering assignment for each point in the dataset, it proceeds in two stages (see Algorithm 3 for details):

- Stage 1: In-sample clustering
  - a Draw a subset $\hat{\mathbf{Y}}$ of $n \ll N$ points. Step 1 in Algorithm 3.
  - b For each point $\hat{\mathbf{y}}_i \in \hat{\mathbf{Y}}$, find its $d_{\max}$ nearest neighboring points in $\mathbf{Y}$ and use $\mathcal{C}_i$ to denote the index set of these points. We call $\mathbf{Y}_{\mathcal{C}_i}$ the sub-cluster corresponds to $\hat{\mathbf{y}}_i$. Step 2 in Algorithm 3.
  - c Compute the affinity matrix $\mathbf{D}$ where each element $\mathbf{D}_{ij}$ is the similarity calculated between $\mathbf{Y}_{\mathcal{C}_i}$ and $\mathbf{Y}_{\mathcal{C}_j}$. Step 3 in Algorithm 3.
  - d Sparsify the affinity matrix by removing possible spurious connections. Step 4 in Algorithm 3.
  - e Cluster the points in $\hat{\mathbf{Y}}$ based on spectral clustering of the sparsified affinity matrix. Step 5 in Algorithm 3.
- Stage 2: Out-of-sample classification

a Fit a classifier to the clustered points in the subset and classify the out-of-sample points in $\mathbf{Y} \setminus \hat{\mathbf{Y}}$. This is Step 6 in Algorithm 3.

Step (1b) computes a neighborhood of points around each sampled points by thresholding inner product similarities, the same method that was used in Heckel and Bölcskei [2015]. The intuition behind this step is that for normalized data, two vectors are more likely to lie in the same linear subspace if the absolute magnitude of the inner product between the points is large. We may also use other measure of similarities in Step (1b) to find the neighboring points. In Section 4.4 and Appendix C.3, we will present the experimental results based on other measure of similarities as well. More generally, for non-linear clustering problems, we can use kernels to replace inner products in measuring similarities.

The idea of using distance between the sub-clusters to construct an affinity matrix in step (1c) relies on the self-representative property of linear subspaces — see Theorem 4.3.2 for technical details as well as some of the basic concepts underlying self-representation. Please note that each entry of affinity matrix measures the closeness between data points, hence it decreases with distance function. There is both theoretical and empirical evidence that sparsification of an affinity matrix by setting smaller elements to zero improves clustering results [Belkin and Niyogi, 2003, Von Luxburg, 2007]. For this reason in step (1d) we threshold the affinity matrix. Once the subset is clustered, the remaining points are labeled via a regression approach where a regression model is fitted on the clustered data, specifically a residual minimization model by ridge regression. Note that any classifier can be used to do the out-of-sample classification. While ridge regression model is proved to work well for linear subspace clustering problems in this chapter, we encourage users of Algorithm 3 to choose different classifier (svm, random forest, or even deep neural networks etc.) based on their own understandings of data.

4.2.2 Practical Recommendations for Parameter Setting

In Algorithm 3 (SBSC), we assume the number of clusters is known—several methods have been developed for the estimation of the number of clusters from data, see Ng et al. [2002]. Intuitively, n should be large enough so that it can well represent the structure of whole data set while still be relatively small to reduce the computational complexity, in our numerical experiments, we choose n to be linear in $K \log N$. See Section 4.3 for theoretical considerations.

input : Data $\mathbf{Y}$, number of subspaces K, sampling size n, neighbor threshold $d_{\max}$, regularization parameters λ_1 and λ_2, residual minimization parameter m, affinity threshold $t_{\max}$.

output : The label vector $\boldsymbol{\ell}$ of all points in $\mathbf{Y}$

1. Uniformly sample n points $\hat{\mathbf{Y}}$ from $\mathbf{Y}$.

2. Construct the sub-clusters:

for *i = 1* **to** n **do**
  $\mathbf{p} = |\langle \hat{\mathbf{y}}_i, \mathbf{Y} \rangle|$;
  $\mathcal{C}_i := \{j : |\langle \hat{\mathbf{y}}_i, \mathbf{y}_j \rangle| \geq \mathbf{p}_{(N-d_{\max})}\}$.
end

3. Construct affinity matrix $\mathbf{D}_{ij} = e^{-d(\mathbf{Y}_{\mathcal{C}_i}, \mathbf{Y}_{\mathcal{C}_j})/2}$ for $i \neq j \in \{1, ..., n\}$ and

$$\begin{aligned} d(\mathbf{Y}_{\mathcal{C}_i}, \mathbf{Y}_{\mathcal{C}_j}) &= ||\mathbf{Y}_{\mathcal{C}_i} - \mathbf{Y}_{\mathcal{C}_j}(\mathbf{Y}_{\mathcal{C}_j}^T \mathbf{Y}_{\mathcal{C}_j} + \lambda_1 \mathbf{I})^{-1} \mathbf{Y}_{\mathcal{C}_j}^T \mathbf{Y}_{\mathcal{C}_i}||_F \\ &\quad + ||\mathbf{Y}_{\mathcal{C}_j} - \mathbf{Y}_{\mathcal{C}_i}(\mathbf{Y}_{\mathcal{C}_i}^T \mathbf{Y}_{\mathcal{C}_i} + \lambda_1 \mathbf{I})^{-1} \mathbf{Y}_{\mathcal{C}_i}^T \mathbf{Y}_{\mathcal{C}_j}||_F. \end{aligned}$$

4. Sparsify the adjacency matrix:

for $i = 1$ **to** n **do**
  $\mathbf{v} := \mathbf{D}_{i\cdot}$;
  for $j = 1$ **to** n **do**
    if $\mathbf{D}_{ij} \leq \mathbf{v}_{(n-t_{\max})}$ **then**
      $\mathbf{D}_{ij} := 0$
    end
  end
end

5. Cluster $\hat{\mathbf{Y}}$: let $\widetilde{\mathbf{D}} = \mathbf{D} + \mathbf{D}^T$ and cluster the in sample points in $\hat{\mathbf{Y}}$ by applying spectral clustering on $\widetilde{\mathbf{D}}$, use $\boldsymbol{\ell}_{in}$ to denote the labels of $\hat{\mathbf{Y}}$.

6. Label the remaining points: use the Residual Minimization by Ridge Regression (RMRR) algorithm in Appendix C.2 to classify the remaining points in $\mathbf{Y} \setminus \hat{\mathbf{Y}}$, specifically for the out-of-sample label we have

$$\boldsymbol{\ell}_{out} = RMRR(\mathbf{Y} \setminus \hat{\mathbf{Y}}, \hat{\mathbf{Y}}, \boldsymbol{\ell}_{in}, \lambda_2, m)$$

7. Combine $\boldsymbol{\ell}_{in}$ and $\boldsymbol{\ell}_{out}$ to get $\boldsymbol{\ell}$, the label of the whole dataset $\mathbf{Y}$.

Algorithm 3: Sampling Based Subspace Clustering (SBSC) algorithm.

Ideally, each sub-cluster $\mathbf{Y}_{\mathcal{C}_i}$ should well represent the subspace it belongs to, i.e. contains at least one basis of that subspace. Therefore we want $d_{\max}$ to be larger than $\max_{k=1,\dots,K} d_k$ which is unknown, for this reason we set $d_{\max}$ to be linear in D. Similarly the residual minimization parameter m should also be linear D.

In choosing λ_1, we recommend using $\lambda_1 = \frac{1}{D}(\max_{i=1,\dots,n} \sqrt{\sum_{j=1}^{d} \frac{1}{a_{ij}^2}})^{-1}$ here a_{ij} corresponds to the j-th positive singular value of $\mathbf{Y}_{\mathcal{C}_i}\mathbf{Y}_{\mathcal{C}_i}^T$, see Appendix C.1 for analytical considerations.

Threshold Selection The spectral clustering algorithm can deliver exact clustering result [Von Luxburg, 2007] if the graph induced by the affinity matrix $\mathbf{D} + \mathbf{D}^T$ has no false connections; and has exactly K connected components. For a large threshold parameter $t_{\max}$ on the affinity matrix more entries in $\mathbf{D}$ will be kept and our algorithm is more likely to have false connections, while small $t_{\max}$ eliminates false connections but might incur non-connectivity.

Let us consider a heuristic situation: the subset we sampled contains exactly the same points (hence $\frac{n}{K}$ points) for each cluster. Then if we choose the threshold index $t_{\max}$ to be $\frac{n}{2K}$, the induced graph from our affinity matrix will have no false connection (given that points from same subspace have bigger similarities between each other) and the clusters themselves will be connected, therefore the spectral clustering algorithm will deliver the exact clustering result [Luxburg et al., 2005].

In reality clusters do not usually have same points in $\hat{\mathbf{Y}}$, hence we choose $t_{\max}$ to start from a relatively large number $\frac{n}{0.5K}$ and gradually increase it. Based on different threshold values, we can generate different label vectors on the subset $\hat{\mathbf{Y}}$, intuitively label vectors that can deliver highly accurate results should be similar to each other or stable. Based on this intuition, we developed a simple adaptive algorithm for finding an "optimal" affinity threshold $t_{\max}$, see supplementary code for details. Based on our observation, choosing t_{max} adaptively works well with datasets where each cluster has large amount of points.

Combining Runs of the Algorithm Thanks to the speed efficiency of our algorithm. We can conduct several independent runs for one experiment (for sampling based algorithms, the results between independent runs might be different) with decent running time. In order to make full use of such advantage, we designed an algorithm to combine the results, or "bagging" the results, from several runs of SBSC. Unlike the classification problem, we need to unify the label vectors before

voting or in other words we need to deal with label switching, see the supplementary code for details on how label switching is addressed. Please note that this bagging algorithm can be used for any clustering algorithms. In Section 4.4 and Appendix C.3 we report the results, both with and without bagging, for all sampling based algorithms.

4.2.3 Comments on the Algorithm

In this section, we make some comments on SBSC to explain the intuitions behind it.

Motivation of Sampling A theoretical result was developed in Luxburg et al. [2005], where under certain assumptions, the spectral clustering results on subset $\hat{\mathbf{Y}}$ will converge to the results on the whole dataset $\mathbf{Y}$. While the result is not directly applicable to our algorithm since it requires the distance function to be continuous and larger than a fixed constant, it gives us the insight that as the sample size n increases properly with N, $\hat{\mathbf{Y}}$ is almost as informative as $\mathbf{Y}$.

Another motivation of using sampling based algorithm is the computational limitation. Traditional spectral clustering based algorithms need to build the "neighborhood" for each of the N points (by lasso, omp etc.), thus the complexity is usually at least $O(N^2)$ (both in time and space), while sampling based algorithms do this step only for the subset, using classification algorithms to label the out-of-sample points requires $O(N \log N)$ in time (given that n, D are linear in $\log N$) with much less memory.

Advantages over Existing Sampling Based Methods While most sampling based algorithms use only the information in $\hat{\mathbf{Y}}$, our algorithm seeks to borrow information from $\mathbf{Y}$ by finding nearest points for each sampled point among the whole dataset. This makes it possible to get a neighborhood with decent size and no false connections for each sampled point. While for traditional methods that apply clustering algorithms purely on the subset, each sampled point only has few neighboring points.

The affinity matrix we build on $\hat{\mathbf{Y}}$ is calculated from the sub-cluster-wise distance, under which the affinity between two points in $\hat{\mathbf{Y}}$ is measured by the affinity between the corresponding sub-clusters. In this chapter, we empirically demonstrated the advantages of SBSC over existing sampling based methods by showing that clustering through sub-clusters can significantly boost the clustering accuracy on the subset.

4.3 Clustering Accuracy

In this section, we analyze several theoretical properties of SBSC. Specifically, we proved that under certain conditions, our algorithm has sub-cluster preserving property (defined later) in Stage 1 and can deliver exact out-of-sample classification in Stage 2, all with high probabilities. Throughout this section we conducted our analysis under the noisy case, for simplicity we also assume all subspaces have same dimension d. The clustering problem is hard with large $\frac{d}{N}$ ratio, since this means each point is less informative.

4.3.1 Model Specification for Provable Results

Note that in Algorithm 3 we assume the data matrix $\mathbf{Y}$ has unit column norm, this can always be achieved by normalizing each column of original data matrix. Specifically, we write the data generating equation for the original data point as

$$\hat{\mathbf{y}_\mathbf{i}}^{(k)} = \zeta_i^{(k)} \mathbf{U_k} \mathbf{a}_i^{(k)} + \hat{\mathbf{e}}_i^{(k)},$$

here $\mathbf{a}_i^{(k)} \in \mathbb{R}^d$ is sampled from the uniform distribution on the surface of $\mathbb{S}^{d-1}$, $\zeta_i^{(k)}$ is a random scalar such that $\zeta_i^{(k)2} \sim \chi_d^2$, and $\hat{\mathbf{e}}_i^{(k)} \sim \mathcal{N}(\mathbf{0}, d\sigma^2 \mathbf{I}_D)$.

Write $\mathbf{y}_i^{(k)}$ as the normalized version of $\hat{\mathbf{y}_\mathbf{i}}^{(k)}$. We have the following relation

$$\mathbf{y}_i^{(k)} = \frac{\mathbf{U_k} \mathbf{a}_i^{(k)} + \sigma \mathbf{e}_i^{(k)}}{||\mathbf{U}_k \mathbf{a}_i^{(k)} + \sigma \mathbf{e}_i^{(k)}||_2}. \tag{4.1}$$

One can show that each entry in $\mathbf{e}_i^{(k)}$ follows t-distribution with d degrees of freedom, and $\frac{||\mathbf{e}_i^{(k)}||_2^2}{D} \sim F_{D,d}$ (F-distribution with parameters (D, d)). Numerically, the normalizing constant $||\mathbf{U}_k \mathbf{a}_i^{(k)} + \sigma \mathbf{e}_i^{(k)}||_2$ will be approximately 1. In Heckel and Bölcskei [2015], the normalizing constants are treated directly as 1, under which $\mathbf{e}_i^{(k)}$ is a multivariate Gaussian vector. In Section 4.3, we explicitly account for the normalizing constant $||\mathbf{U}_k \mathbf{a}_i^{(k)} + \sigma \mathbf{e}_i^{(k)}||_2$ and have done detailed distributional analysis.

We write $\lambda_1^{(ij)} \geq \lambda_2^{(ij)} \geq ... \geq \lambda_d^{(ij)}$ correspond to the *cosine* values of principal angles between $\mathcal{S}_i$ and $\mathcal{S}_j$, hence $\lambda_1^{(ij)} \leq 1$ and $\lambda_d^{(ij)} \geq 0$. Note $\lambda_k^{(ij)} = \lambda_k^{(ji)}$ for $1 \leq k \leq d$ and $1 \leq i < j \leq K$. For each subspace $\mathcal{S}_k$, we define the uniformly maximal affinity vector to quantify its closeness with respect to all other subspaces.

Definition 4.3.1. *For each subspace $\mathcal{S}_k$, its uniformly maximal affinity vector with respect to other*

subspaces is $[\lambda_1^{(k)}, ..., \lambda_d^{(k)}]$ *such that*

$$\lambda_i^{(k)} = \max_{j \neq k} \lambda_i^{(kj)}.$$

If the uniformly maximal affinity vectors have small entries, we should be able to decrease the "false discovery" in $\{\mathbf{Y}_{\mathcal{C}_i}\}_{i=1}^n$. Formally, we have the following definition.

Definition 4.3.2. *We say Algorithm 3 has sub-cluster preserving property if each* $\mathbf{Y}_{\mathcal{C}_i}$ *only contains points from the same subspace as* $\hat{\mathbf{y}}_i$.

If SBSC has sub-cluster preserving property and $\mathbf{Y}_{\mathcal{C}_i}$ concentrates around $\mathcal{S}_k$, we can write $\mathbf{Y}_{\mathcal{C}_i} = \mathbf{U}_k\hat{\mathbf{B}}_i + \hat{\mathbf{E}}_i$. Here each column of $\hat{\mathbf{B}}_i \in \mathbb{R}^{d\times(d_{\max}+1)}$ is a sample from uniform distribution on $\mathbb{S}^{d-1}$ divided by its corresponding normalizing constant, and each column of $\hat{\mathbf{E}}_i \in \mathbb{R}^{D\times(d_{\max}+1)}$ is a noise vector divided by its corresponding normalizing constant. We write $\hat{\mathbf{B}}_{i'j}$ as the j-th column of matrix $\hat{\mathbf{B}}_i$, and similarly for $\hat{\mathbf{E}}_{i'j}$. Then we have $\|\mathbf{U}_k\hat{\mathbf{B}}_{i'j} + \hat{\mathbf{E}}_{i'j}\|_2 = 1$. For convenience we also write $\mathbf{B}_i$ as the "un-normalized" version of $\hat{\mathbf{B}}_i$, hence $\hat{\mathbf{B}}_i$ has unit column norm, similar notation is used for $\mathbf{E}_i$.

In constructing the affinity matrix $\mathbf{D}$, we want the following property: points that belong same subspace have relatively bigger affinities (hence smaller distances) between each other, this property can be formally defined as:

Definition 4.3.3. *We say* $\mathbf{Y}_{\mathcal{C}_i}$ *has the correct neighborhood property with distance function* $d(\cdot,\cdot)$ *if*

$$d(\mathbf{Y}_{\mathcal{C}_i}, \mathbf{Y}_{\mathcal{C}_j}) < d(\mathbf{Y}_{\mathcal{C}_i}, \mathbf{Y}_{\mathcal{C}_k})$$

for any $1 \leq j \neq k \leq n$ *such that* $\mathbf{Y}_{\mathcal{C}_j}$ *concentrates around the same subspace with* $\mathbf{Y}_{\mathcal{C}_i}$ *and* $\mathbf{Y}_{\mathcal{C}_k}$ *concentrates around a different subspace.*

4.3.2 Theoretical Properties of SBSC

In this section, we will discuss three theoretical properties regarding Algorithm 3, detailed proofs can be found in Appendix C.1.

Assumptions In this section, we list all the assumptions used by lemmas and theorems of this chapter, notice $A2$ subsumes $A1$ part 2, $A3$ includes $A1$, and $A4$ assumes a modification of $A1$, $A2$ and $A3$.

A1. There exist constants T_l, T_u, and ρ such that:

$$T_l^2 \leq \min_{k=1,\ldots,K} Q_{1-\frac{d_{max}}{N_k^{1-\rho}}}, \quad T_l \in (0, \frac{\sqrt{2}}{2}), \tag{4.2}$$

$$T_u^2 \geq \max_{k=1,\ldots,K} Q_{1-\frac{1}{N_k^{1+\rho}}}, \quad T_u \in (0, \frac{\sqrt{2}}{2}), \tag{4.3}$$

here Q_p denotes the upper p quantile of a Beta distribution with parameters $(\frac{1}{2}, \frac{d-1}{2})$.

A2. There exist positive constants $\{g_i\}_{i=1}^2$, η and $\rho \in (0,1)$, such that if we write $T = \frac{4g_2+2g_2^2}{1-g_2} + \frac{1+g_2}{1-g_2} g_1$, the following inequalities hold: (4.2) with T_l replaced by g_1, and

$$\sum_{i=1}^{d} \left(g_1^2 - \lambda_i^{(k)2}\right)_+^2 > \sum_{i=1}^{d} \left(g_1^2 - \lambda_i^{(k)2}\right)_-^2, \quad \sum_{i=1}^{d} \left(g_1^2 - \lambda_i^{(k)2}\right)_+ > \sum_{i=1}^{d} \left(g_1^2 - \lambda_i^{(k)2}\right)_-, \tag{4.4}$$

$$\frac{g_2^2}{D\sigma^2} > 3 + \frac{6}{\eta}, \quad \frac{d}{\log N} \geq (2+2\eta)^2. \tag{4.5}$$

A3. There exist positive constants T_l, T_u, q_0, ρ and t such that the following inequalities hold: (4.2), (4.3), $d_{max} > d$ and

$$\frac{(T_l^2 d_{max} - C_2)C_2 - T_u^2(1-T_u^2)C_1^2}{T_l^2 d_{max}} \geq q_0, \tag{4.6}$$

here

$$C_1 = \left(2 + t\sqrt{\frac{\log N}{d-2}}\right)\sqrt{d_{max}}, \; C_2 = \left(1 - T_u^2\right)\left(\sqrt{\frac{2d_{max}}{\pi(d-1)}} - 2 - t\sqrt{\frac{\log N}{d-2}}\right)^2.$$

A4. There exist positive constants T_l, T_u, g, λ, η, q_0, ρ and t such that the following inequalities hold: (4.2), (4.3), (4.4) with g_1 replaced by T_l, (4.5) with g_2 replaced by g, (4.6) and

$$\frac{(2g-g^2)(1+g)\sqrt{d}\,(d_{max}+1)}{q_0(1-g)} \leq \frac{1}{2}, \; \frac{5\lambda(1+g)^2\sqrt{d(d_{max}+1)}}{q_0(1-g)} \leq \sqrt{1-T_l^2}. \tag{4.7}$$

Assumption $A1$ is used to bound the order statistics of a Beta distribution in Lemma C.1.2. For example, if we write $N_k = 10000$, $d_{max} = 3d$, $T_l^2 = 0.08$, and $\rho = 0.05$, then inequality (4.2) translates to $\frac{d}{\log N_k} < 6$. With the same setting and we pick $T_u^2 = 0.4$, then (4.3) translates approximately to

$\frac{d}{\log N_k} \geq 4$. In general, the bounds on $\frac{d}{\log N_k}$ are wider for larger N_k.

Assumption $A2$ is the subspace separation assumption. We use it for the proof of Theorem 4.3.1. In Appendix C.1, we show that SBSC requires most of $\{\lambda_i^{(k)}\}_{i=1}^{d}{}_{k=1}^{K}$ to be smaller than g_1. This means large g_1 implies an easier clustering problem for SBSC, and vice versa. Throughout this chapter we call g_1 the affinity threshold. Note that T is a upper bound of the affinity threshold g_1, specifically if there was no noise $T = g_1$. From (4.2) we know that large $\frac{d}{\log N}$ implies a small T and g_1. Therefore, large d makes the clustering problem harder. This agrees with our intuition. Consider the extreme case where the subspaces are orthogonal with each other, then $\{\lambda_i^{(k)}\}_{i=1,k=1}^{d,\ K}$ are 0. This means (4.4) are naturally true with any positive constant g_1. Finally, the constant g_2 in (4.5) controls the noise term. From the first condition in (4.5) we have $\sigma < \frac{g_2}{\sqrt{D}}$.

Assumption $A3$ guarantees the sub-clusters $\{\mathbf{Y}_{\mathcal{C}_i}\}_{i=1}^{n}$ are informative. We use it mainly for the proof of Lemma C.1.5. Here the term C_2 is closely related to the permeance statistics (Lerman et al., 2012), which measures how well a set of vectors is scattered across a space. Therefore a large $\frac{d_{max}}{d}$ implies that these vectors are well scattered. Specifically, if $\rho = 0.05$, $N_k = 100000$, $d_{max} = 90d$, and we want $q_0 \geq 1$, $A3$ requires $\frac{d}{\log N_k} \leq 5.5$ [1].

Assumption $A4$ is a combination of all previous assumptions, with slightly stronger conditions on subspace similarities and noise level; we use it for the proof of Theorem 4.3.2. Here g again controls the magnitude of the norm of noise terms.

Theoretical Properties of SBSC Two theorems regarding the State 1 of SBSC are discussed in this section.

Theorem 4.3.1. *Under Assumption A2, SBSC has sub-cluster preserving property with probability at least*

$$1 - \sum_{j=1}^{K} \frac{n_j(N_j - d_{\max})}{d_{\max}(N_j + 1)(N_j^{\rho} - 1)^2} - 2(K-1)ne^{-\epsilon^2} - \frac{2N}{N^{(1+\frac{\eta}{2+\eta})^2} - 2}, \tag{4.8}$$

[1] As we change $\frac{d}{\log N_k}$ from 4 to 5.5, T_l^2 changes from 0.06 to 0.04, T_u^2 changes from 0.36 to 0.28. In this example, $\frac{d_{max}}{d}$ is fairly large. In the numerical section we found it is usually not necessary to choose large d_{max}. A better bound in Corollary C.1.3 might be helpful the bridge the gap between numerical experiments and theoretical guarantee.

where

$$\epsilon = \min_k \frac{\sum_{i=1}^d \left(g_1^2 - \lambda_i^{(k)2}\right)_+ - \sum_{i=1}^d \left(g_1^2 - \lambda_i^{(k)2}\right)_-}{2\sqrt{\sum_{i=1}^d \left(g_1^2 - \lambda_i^{(k)2}\right)_+^2} + \sqrt{4\sum_{i=1}^d \left(g_1^2 - \lambda_i^{(k)2}\right)_+^2 + 2\sum_{i=1}^d \left(g_1^2 - \lambda_i^{(k)2}\right)_+}}. \tag{4.9}$$

If the subspaces are orthogonal with each other, i.e. $\{\lambda_i^{(k)}\}_{i=1,k=1}^{d,\ K} = 0$. Equation (4.9) is

$$\epsilon = \frac{\sqrt{d}}{2 + \sqrt{4 + \frac{2}{g_1^2}}}.$$

This shows ϵ is linear in $\sqrt{d}$ and monotonically increasing in g_1^2. Appendix C.4 establishes general conditions on g_1 and $\{\lambda_i^{(k)}\}_{i=1,k=1}^{d,\ K}$ under which ϵ grows like $\sqrt{d}$. Combining this with Assumption $A2$, we observe that the third term of (4.8) is small for large N.

Next, we use the sub-cluster preserving property established in Theorem 4.3.1 to prove the theoretical guarantee for correct neighborhood property (see Definition 4.3.3). We define a distance function between two sub-clusters as

$$d(\mathbf{Y}_{\mathcal{C}_i}, \mathbf{Y}_{\mathcal{C}_j}) = ||\mathbf{Y}_{\mathcal{C}_i} - \mathbf{Y}_{\mathcal{C}_j}(\mathbf{Y}_{\mathcal{C}_j}^T\mathbf{Y}_{\mathcal{C}_j} + \lambda\mathbf{I})^{-1}\mathbf{Y}_{\mathcal{C}_j}^T\mathbf{Y}_{\mathcal{C}_i}||_F + ||\mathbf{Y}_{\mathcal{C}_j} - \mathbf{Y}_{\mathcal{C}_i}(\mathbf{Y}_{\mathcal{C}_i}^T\mathbf{Y}_{\mathcal{C}_i} + \lambda\mathbf{I})^{-1}\mathbf{Y}_{\mathcal{C}_i}^T\mathbf{Y}_{\mathcal{C}_j}||_F, \tag{4.10}$$

here $\lambda > 0$ is the regularization parameter.

Theorem 4.3.2. *Assume sub-cluster preserving property is true for SBSC with probability at least $1 - p_s$, and Assumption A4 is satisfied. Then $\{\mathbf{Y}_{\mathcal{C}_i}\}_{i=1}^n$ have the correct neighborhood property with the distance function* (4.10) *with probability at least*

$$1 - p_s - 4n(n-1)e^{-\epsilon^2} - \frac{2n}{N^{t^2/2}} - \sum_{k=1}^K \frac{2n_k(N_k - d_{max})}{d_{max}(N_k+1)(N_k^\rho - 1)^2} - \frac{2N}{N^{(1+\frac{\eta}{2+\eta})^2} - 2},$$

here ϵ is defined in (4.9) *with g_1 replaced by T_l.*

4.4 Experimental Results

In this section, we test our algorithm on both synthetic and real world data sets. In addition to SBSC, we also tried using different methods to find neighboring points in Step 2 of SBSC. For the rest of this section and Appendix C.3, we call the original SBSC algorithm (described in Algorithm 3)

as SBSC-TSC, the algorithm that replaces Step 2 with lasso minimization to find neiboring points in Algorithm 3 as SBSC-SSC, and similarity for SBSC-DSC.

The performance of SBSC family is compared to other state-of-the art algorithms. This include classic subspace clustering method: Sparse Subspace Clustering (SSC,Elhamifar and Vidal, 2009, You et al., 2016a), Thresholding Subspace Clustering (TSC, Heckel and Bölcskei, 2015), Direction Search Subspace Clustering (DSC, Rahmani and Atia, 2017), Least Squares Regression (LSR, Lu et al., 2012), Low-Rank Representation (LRR, Liu et al., 2010), Subspace Clustering by Orthogonal Matching Pursuit (SSC-OMP, You et al., 2016c), Elastic Net Subspace Clustering (ENSC, You et al., 2016b); and scalable representation-based algorithms: Scalable Sparse Subspace Clustering (SSSC), similarly for other notations like STSC, SDSC etc. [Peng et al., 2013]. For most of them we replicated their results on our machine to make fair comparisons, some of the results were copied from the original papers due to the unavailability of codes.

Throughout this section, we use clustering accuracy [You et al., 2016c], nmi [Zhou et al., 2018] and running time as the metrics for performance evaluation. To demonstrate the advantages of using sub-clusters (i.e. borrowing information from the whole dataset) to cluster the data points in the subset, for sampling based algorithms we also report their clustering accuracy on the subset. In the rest of this chapter, we call the clustering accuracy on the whole data set as accuracy, and the clustering accuracy on the subset as accuracy-sub. For randomized algorithms, reported results are averaged over 10 trials. The parameters setup for all algorithms can be found directly in the supplementary codes.

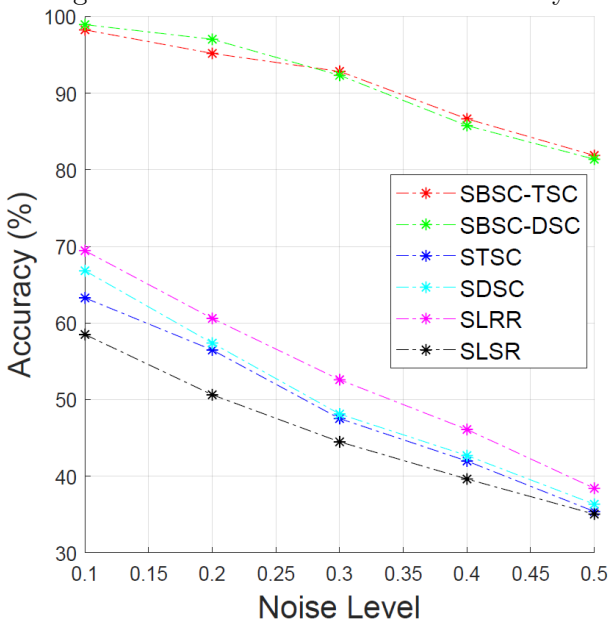
Note that due to limitations of running time and space, for different datasets we might present results on different sets of algorithms. A more comprehensive report on numerical results is presented in Appendix C.

4.4.1 Results on Synthetic Data Set

The data generation mechanism of our synthetic data was based on Section 4.3.1. For synthetic datasets, we want to compare the tolerance to noise between different sampling based algorithms as well as the scalability of our algorithm.

Tolerance to Noise In this section, we test the tolerance to noise of our algorithm. From (4.1) we can calculate $\mathbb{E}\left[\sigma^2\|\mathbf{e}_i^{(k)}\|_2^2\right] = \sigma^2\frac{Dd}{d-2}$. Since the un-normalized signal $\mathbf{U}_k\mathbf{a}_i^{(k)}$ of $\mathbf{y}_i^{(k)}$ has unit

Figure 4.1: Tolerance to Noise: Accuracy

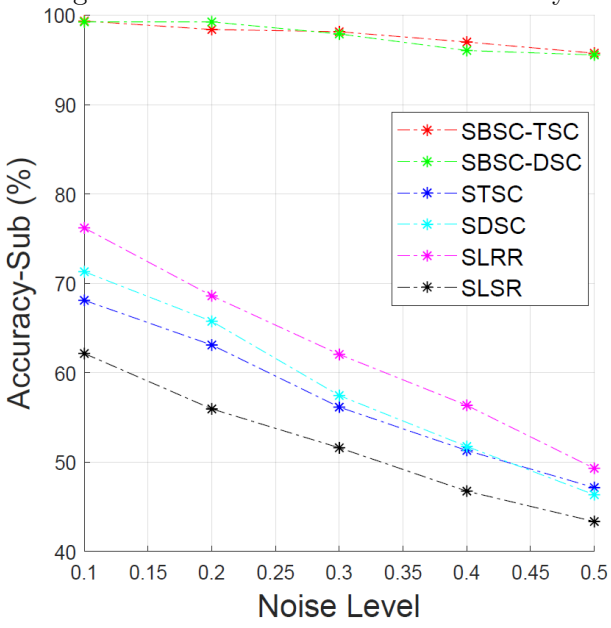


norm, throughout this chapter we define $\sqrt{\frac{Dd}{d-2}}\sigma$ as the noise level.

We change the noise level from 0.1 to 0.5. For each noise level, we simulate 10 datasets, each of them has $K = 20$ subspaces, where each subspace contains $N_i = 10000$ data points. The result is averaged over 10 different datasets for each noise level.For all the sampling based algorithms we fixed $n = 200$ as the sampling size.

The results are presented in Figure 4.1 (accuracy) and Figure 4.2 (accuracy-sub). The small discrepancy between two sides shows both sampling based algorithms can deliver consistent results between in sample clustering and out-of-sample classification. At the same time, the SBSC based algorithms constantly deliver much higher accuracy-sub, this means for the synthetic datasets, borrowing information from the whole data set significantly enhanced the clustering results for subset.

Figure 4.2: Tolerance to Noise: Accuracy



Scalability In this section, we test the scalability of SBSC-TSC. [2] Specifically, we randomly generate $K = 20$ subspaces in an ambient space with dimension $D = 30$, each of the subspaces has dimension $d_i = 5$. We increase N_i from 100 to 51200, so the corresponding N increases from 2000 to 1024000. The sampling size n is $\lfloor 2K \log(N) \rfloor$.

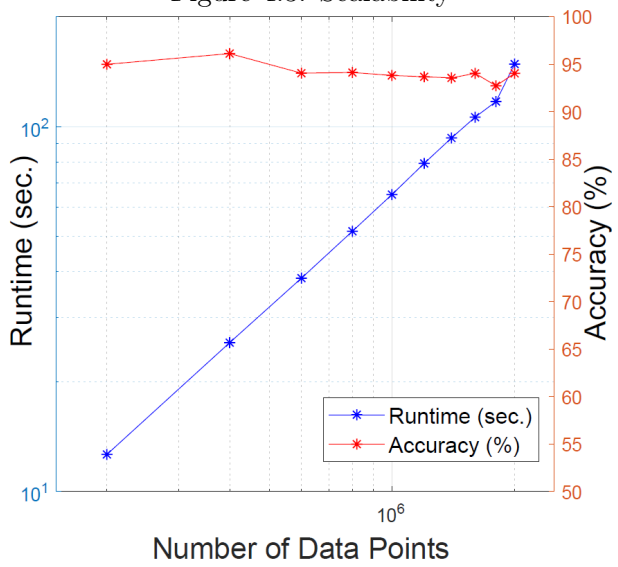
The result is presented in Figure 4.3. On the right hand side y-axis, we show the average accuracy, which is around 95% across all experiments, against the number of data points N, this could justify our choice of n. On the left hand side y-axis, we show the scale plot between running time and N, the linear pattern here agrees with our complexity analysis. As we increase the number of data points N, the accuracy on the whole data set slightly gets higher, this implies our algorithm is particularly useful for large datasets.

4.4.2 Results on Real World Datasets

In this section, we test SBSC on three real world datasets. These datasets were selected to have small, medium and large data size respectively. As expected, the advantage of SBSC over other

[2] The SBSC-SSC and SBSC-DSC algorithms run fairly slow under our simulation setting. The main reason is, we only tried using vanilla solvers to handle the additional optimization problems in SBSC-SSC and SBSC-DSC, which could be inefficient. Existing accelerated version of SSC [You et al., 2016a] is designed for the full data matrix, and DSC solver also involves manipulation of full data matrix. Exploring scalable SSC and DSC solvers for partial data matrix beyond the scope of this chapter.

Figure 4.3: Scalability



We change the number of points N from 2000 to 1024000, for each N we generate 10 datasets and the final result is averaged over 10 independent runs.

state-of-the-art algorithms changes from marginal to significant.

The Extended Yale B dataset The Extended Yale B dataset (YaleB) contains $N = 2432$ face images of $K = 38$ individuals. Each image is a front view photo of the corresponding individual with different illumination condition. To speed up the running time, a dimension reduction step is taken to pre-process the dataset (see Rahmani and Atia [2017]), hence in our experiment $D = 500$.

The result is summarized in Table 4.1. The first six algorithms are sampling based algorithms, the sampling sizes are summarized in the supplementary codes. We see the performance of DSC is superior among all comparing algorithms. As expected, SBSC does not do well for this small dataset. For small dataset, the information loss caused by sampling is relatively large compare to the information contained in the whole dataset, especially with such large number of clusters.

Method	**Accuracy (%)**	**Accuracy-Sub (%)**	**NMI (%)**	**Runtime (sec.)**
SBSC-TSC	26.53 (1.8)	31.56 (1.94)	41.67 (1.44)	26.71
SBSC-DSC	60.46 (1.62)	62.76 (1.88)	70.15 (0.97)	34.83

STSC	17.45 (1.27)	21.72 (1.7)	29.17 (1.28)	0.83
SDSC	52.18 (1.96)	60.78 (2.04)	54.61 (1.85)	2.92
SLRR	18.35 (0.64)	28.6 (1.64)	26.33 (0.57)	6.94
SLSR	26.48 (1.98)	37.78 (2.33)	35.21 (2.22)	1.47
TSC	26.19	NA	39.31	0.89
DSC	**91.69**	NA	**93.43**	**44.79**
SSC	52.96	NA	60.15	169.46
SSC-OMP	67.63	NA	77.03	0.91
SSC-ENSC	60.81	NA	69.4	3.1

Table 4.1: Results on Extended Yale B

The Zipcode dataset The Zipcode dataset is a medium-sized dataset with $N = 9298$ data points and $D = 256$, each point represents an image of handwritten digit, hence $K = 10$.

The result is summarized in Table 4.2. Here "NA" means not available, similarly for next table. For Zipcode, SBSC-TSC delivers the best results in all metrics except running time. But differences in running time are marginal for sampling based algorithms in this medium-sized dataset. The accuracy-sub of SBSC is again better than that of traditional sampling based algorithms (see SBSC-TSC versus STSC, and SBSC-DSC versus SDSC).

In summary, for medium-sized data set, the information loss no longer causes significant deficiency in clustering accuracy. Additionally, compared to the traditional methods like TSC, SSC and DSC, the computational advantage of sampling based algorithms becomes more obvious than the results of small dataset.

Method	**Accuracy (%)**	**Accuracy-Sub (%)**	**NMI (%)**	**Runtime (sec.)**

SBSC-TSC	**69.4 (5.17)**	**72.04 (5.37)**	**70.3 (1.75)**	**10.4**
SBSC-DSC	60.84 (2.87)	64.92 (3.37)	62.92 (0.65)	71.25
STSC	55.28 (4.25)	60.86 (3.8)	53.1 (2.49)	2.4
SDSC	45.62 (6.43)	51.16 (7.31)	45.99 (3.88)	3.155
SLRR	63.21 (3.96)	65.16 (4.03)	66.09 (1.39)	10.14
SLSR	58.66 (0.99)	59.85 (0.98)	62.54 (1.38)	4.14
TSC	65.73	NA	78.97	115.18
DSC	60.92	NA	68.43	799.67
SSC	48.16	NA	52.37	2164.82
SSC-OMP	17.87	NA	7	2.57
SSC-ENSC	44.65	NA	50.08	35.5

Table 4.2: Results on Zipcode

The MNIST dataset The MNIST dataset (MNIST) contains $N = 70000$ data points, each point represents an image of handwritten digit. The original data was transferred into $\mathbb{R}^{500}$ by convolutional neural network and PCA [You et al., 2016c]. Again $K = 10$.

The result is presented in Table 4.3. Here the results of methods with star marks are copied from original paper. The numbers in the parenthesis in first column are number of bagging. For MNIST, SBSC-TSC dominates in nearly every aspect. The large data size of MNIST makes the sampling based algorithms run much faster than traditional methods, therefore we can report the results of them with bagging method [Peng et al., 2016]. Although not reported here, the results of SBSC did not change a lot with different parameter settings.

Method	**Accuracy (%)**	**Accuracy-Sub (%)**	**NMI (%)**	**Runtime (sec.)**
SBSC-TSC(1)	95.74 (0.28)	96.44 (1.14)	89.9 (0.47)	37.8
SBSC-TSC(6)	**97.15 (0.16)**	**95.25 (1.78)**	**92.59 (0.3)**	**261.77**
STSC(1)	30.2 (2.13)	67.8 (3.95)	11.52 (2.12)	28
STSC(6)	40.12 (2.84)	65.23 (2.22)	22.53 (2.36)	179
SLRR(1)	79.5 (1.19)	79.46 (1.3)	79.9 (1.52)	59
SLRR(6)	81 (0.67)	79.6 (0.44)	83.75 (0.74)	342
SLSR(1)	75.06 (6.11)	74.62 (5.99)	76.21 (3.63)	54
SLSR(6)	79.64 (0.85)	76.43 (1.85)	81.24 (0.95)	332
TSC	84.63	NA	87.47	1184
SSC (DC1)*	96.55	NA	NA	5254
SSC (DC2)*	96.1	NA	NA	4390
SSC (DC5)*	94.9	NA	NA	1596
SSC-OMP	81.51	NA	84.45	232
SSC-ENSC	93.79	NA	88.8	500

Table 4.3: Results on MNIST

4.5 Conclusion

To the best of our knowledge, our algorithm is the first sacalable subspace clustering algorithm with theoretical performance guarantee. Empirically, it can deliver accurate clustering result with

high efficiency.

While the idea of subsampling was discussed by other researchers before [Peng et al., 2016], the highlights of this chapter are finding neighborhood points among the whole data set and using cluster-wise distance to cluster points in the subset, in turn this is more robust to sampling bias noise.

In calculating cluster-wise distances and classifying out-of-sample points, ridge regression seems to be the most direct method, please note the algorithm itself it highly flexible, readers are encouraged to try different distance functions, classification methods and even metrics in finding neighborhood points.

APPENDIX A

SUPPLEMENTARY MATERIALS FOR Chapter 2

A.1 Proof of Equation (2.1)

Proof. There are X copies of the target fragments in the second urn. Some of the fragments of interest might not survive during the cutting process, therefore we have $W \leq X$. Define $\{A_i\}_{i=1}^{X}$ as the event that the i-th target fragment survives (i.e. being intact after cutting procedure) and is placed in the third urn. Conditional on U_i, the locations of these U_i cuts are uniformly distributed, therefore $p(A_i) = (1 - f/L)^{U_i}$.

In order for the target fragment to be usable by the detector, it has to be longer than T. If $T \leq f$, the sequences that contain the target fragment are always longer than T, then $q_i(U_i) = p(A_i)$. Otherwise we estimate q_i from a lower bound using the probability of an event $A_i \cup E_i$, where E_i is the event of not having cuts within $T - f$ on either one or the other side of the target sequence (see Figure A.1). Recall that $t_1 = \frac{L-T}{L-f}$, $t_2 = \frac{L-2T+f}{L-f}$, $t_3 = 1 - \frac{f}{L}$. Then by inclusion and exclusion $p(E_i \mid A_i) = 2(t_1)^{U_i} - (t_2)^{U_i}$ and consequently

$$q_i(U_i) \geq p(A_i)p(E_i \mid A_i) = 2(t_1 t_3)^{U_i} - (t_2 t_3)^{U_i}.$$

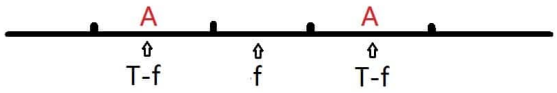


Figure A.1: Demonstration of DNA Cutting
DNA sequence with target fragment. The f zone and at least one of the A zones should have no cuts to provide a valid target sequence.

□

A.2 Hypergeometric Distribution and Binomial Bounds

In this section, we discuss the conditions needed for Theorem 2.3.1. Use the same notation as in Theorem 2.3.1. For fixed positive integer x, consider the following two inequalities

$$P(h = x) \leq P(B_1 = x), \tag{A.1}$$

$$P(h = x) \leq P(B_2 = B - x). \tag{A.2}$$

Note that if (A.1) is true for all $x' \leq x_0$, then (2.4) is true for $x = x_0$, similarly for (A.2). We write $r = \frac{B}{A}$ and expand the above two inequalities as

$$\frac{\binom{A-C}{B-x}}{\binom{A}{B}} \leq r^x(1-r)^{C-x}, \tag{A.3}$$

and

$$\frac{\binom{C}{x}}{\binom{A}{B}} \leq r^{B-x}(1-r)^{A-B-C+x}, \tag{A.4}$$

respectively. We will use (A.3) and (A.4) to derive the conditions needed for Theorem 2.3.1.

Property A.2.1. If (A.3) and (A.4) are true for some fixed A, B, C and $x = x_0 \leq \frac{BC}{A}$. Then Theorem 2.3.1 is true for $x = x_0$.

Proof. We use backward mathematical induction on x to prove (A.3) and (A.4) are true for any $x \leq x_0$. Assume (A.3) and (A.4) are true for some $x = x_0 \leq \frac{BC}{A}$. Then for $x := x_0 - 1$, it suffices to have the following two inequalities

$$\frac{r}{1-r}\frac{A-C-B+x_0}{B-x_0+1} \leq 1, \ \textit{and}\ \frac{1-r}{r}\frac{x_0}{C-x_0+1} \leq 1,$$

which only require $x_0 \leq 1 - r + \frac{BC}{A}$ and $x_0 \leq r + \frac{BC}{A}$, obviously true. Therefore (A.3) and (A.4) are true for any $x \leq x_0$, this means Theorem 2.3.1 is true for $x = x_0$. □

Property A.2.2. Assume (A.3) and (A.4) are true for some fixed A, B, $C = C_0$, and $x = x_0 \leq \frac{BC_0}{A}$. Then Theorem 2.3.1 is true for any $C \in (C_0, A - B)$.

The proof of Property A.2.2 is almost the same as that of Property A.2.1. We present the proof sketch here and the details are omitted: one can fix $x = x_0$ and do forward mathematical induction

on C_0 to show that (A.3) and (A.4) are true for any $C \in (C_0, A-B)$. The property is proved by using Property A.2.1 and the fact that $x_0 \le \frac{BC_0}{A} < \frac{BC}{A}$.

Property A.2.3. Assume the following inequalities are true for some constants A_{up} and B_{low}

$$2A \ge 2B + C, \quad Q \le g\frac{BC}{A}, \quad A \ge \frac{3}{(3-2g)}B + \frac{2}{(3-2g)}C + \frac{3}{(3-2g)}\frac{3A}{C},$$
$$x \ge 5, \quad \psi(A_{up}) \ge 0, \quad (B-Q)C \ge B(2Q+1), \quad A \ge 3B + CG(B_{low}) \ge 0,$$

where $g = \frac{xA}{BC}$ and

$$\begin{aligned}\psi(A) =&(C-1)\log(A) + (C-x)\log(A-B-1) - C\log(A-1) - (C-x-1)\log(A-B)\\ &+ \log(A-C) - \log(A-B-C+x),\\ G(B) =&(x-1)\log(B+1) + (C-x)\log(A-B-1) - x\log B - (C-x-1)\log(A-B)\\ &+ \log(1+B-x) - \log(A-B-C+x).\end{aligned}$$

For fixed B, x and C, if (A.3) is true for $A = A_{up}$, then (2.4) is true for any $A \in (\max\{B, C\}, A_{up})$; for fixed A, x and C, if (A.3) is true for $B = B_{low}$, then (2.5) is true for any $B \in (B_{low}, A)$.

Proof. Using Property A.2.1, we only need to show (A.3) is true accordingly. Again we use (backward) mathematical induction on A. Given $\frac{\binom{A-C}{B-x}}{\binom{A}{B}} \le r^x(1-r)^{C-x}$. We need $\frac{\binom{A-1-C}{B-x}}{\binom{A-1}{B}} \le (\frac{B}{A-1})^x(1-\frac{B}{A-1})^{C-x}$. It suffices to show

$$\frac{\binom{A-1-C}{B-x}}{\binom{A-1}{B}} \le (\frac{B}{A-1})^x(1-\frac{B}{A-1})^{C-x}(\frac{A}{B})^x(1-\frac{B}{A})^{x-C}\frac{\binom{A-C}{B-x}}{\binom{A}{B}}.$$

The inequality above is equivalent to $\psi(A) \ge 0$. Take first order derivative of $\psi(A)$ with respect to A we have

$$\psi'(A) = -\frac{C}{A-1} + \frac{C-1}{A} + \frac{1}{A-C} - \frac{C-x-1}{A-B} + \frac{C-x}{A-B-1} - \frac{1}{A-B-C+x}.$$

If $\psi'(A) \le 0$ for $A \le A_{up}$, the result is proved by using the monotonicity of $\psi(A)$ and the assumption

that $\psi(A_{up}) \geq 0$. Now we will prove $\psi'(A) \leq 0$. It suffices to show

$$\begin{aligned} &- B^3C^2 + B^3C - B^2C^3 + B^2C^2x - B^2Cx + B^2C - BC^3 + BC^2x + BC^2 - BCx \\ &+ A(3B^2C^2 - 3B^2C + 2BC^3 - 2BC^2x + 2BCx - 2BC + C^2x - Cx^2) \\ &+ A^2(-3BC^2 + 3BC - C^2x + Cx^2 - 2Cx + x^2 + x) + A^3(2Cx - x^2 - x) \leq 0. \end{aligned} \tag{A.5}$$

The fist line of (A.5) is obviously negative by noting the following facts

$$B^2C^2x \leq B^2C^3, \quad B^2C \leq B^2Cx, \quad B^3C \leq B^3C^2, \quad BC^2x + BC^2 \leq BC^3 + BCx.$$

For the rest lines, we use the following relations

$$Ax^2 + Ax + ACx^2 \leq 3AB^2C + 2BC^2x, \quad C^2x + 2BCx \leq 2ACx, \quad -x^2 - x \leq 0,$$

where the last inequality follows by assumption $2A \geq 2B + C$. For the rest parts, we want $2xA^2 + 3AB + 2BC^2 + 3B^2C \leq 3ABC$. It suffices to show

$$(3 - 2g)A \geq 3B + 2C + \frac{3A}{C},$$

which is equivalent to

$$A \geq \frac{3}{(3-2g)}B + \frac{2}{(3-2g)}C + \frac{3}{(3-g)}\frac{3A}{C},$$

this follows directly from the assumptions. Thus the first part of Property 3 is proved.

Now we prove the second part. From mathematical reduction on B, we want $\frac{\binom{A-C}{B+1-x}}{\binom{A}{B+1}} \leq (\frac{B+1}{A})^x(1 - \frac{B+1}{A})^{C-x}$. It suffices to have

$$\frac{\binom{A-C}{B+1-x}}{\binom{A}{B+1}} \leq (\frac{B+1}{A})^x(1 - \frac{B+1}{A})^{C-x} \cdot (\frac{A}{B})^x(1 - \frac{B}{A})^{x-C}\frac{\binom{A-C}{B-x}}{\binom{A}{B}},$$

which is equivalent to $G(B) \geq 0$. Similarly as before, we want this function increases with $B \geq B_{low}$, from which we only need to check $G(B_{low}) \geq 0$ and this follows from our assumption. Consider the

first order derivative of $G(B)$

$$G'(B) = \frac{1}{1+B-x} + \frac{C-x-1}{A-B} - \frac{C-x}{A-B-1} + \frac{x-1}{1+B} - \frac{x}{B} + \frac{1}{A-B-C+x}.$$

Then $G'(B) \leq 0$ requires

$$\begin{aligned}
&- B^3(C-1)(C-2x) + B^2C^2(x-2) - BC(x-1) + BC^2(x-1) - 3B^2x + B(x-1)x \\
&- BC(x-1)x + B^2x^2 + B^2C(2+2x-x^2) + A^3(1-x)x \\
&+ A^2\left[(x-1)x + 3B(x-1)x + C(x-1)x + (1-x)x^2\right] \\
&+ A(-3B^2(x-1)x - C(x-1)x - 2BC(x-1)x + 2B(x-1)^2x + (x-1)x^2) \leq 0.
\end{aligned} \tag{A.6}$$

We can expand the first line of (A.6) and write it as

$$\begin{aligned}
&- B^3C^2 + B^3(2x+1)C - 2xB^3 + B^2C^2x - 2B^2C^2 - BCx + BC + BC^2x - BC^2 - 3B^2x \\
&+ Bx^2 - Bx - BCx^2 + BCx + B^2x^2.
\end{aligned}$$

We want to prove that the above line is non-positive. Note that

$$\begin{aligned}
&- BCx + BC \leq 0, \quad BC^2x - 2B^2C^2 \leq 0, \quad -BC^2 \leq 0 \\
&- Bx \leq 0, \quad -3B^2x + Bx^2 \leq 0, \quad -BCx^2 + BCx \leq 0, \quad B^2x^2 - 2xB^3 \leq 0.
\end{aligned}$$

Finally we only need $-B^3C^2 + B^3(2x+1)C + B^2C^2x \leq 0$, which follows from our assumption: $(B-x)C \geq B(2x+1)$.

From $x \geq 5$ we immediately get: $2 + 2x - x^2 \leq 0$, hence $B^2C(2+2x-x^2) \leq 0$. For the second and third terms at the second line of (A.6) we show

$$A(1-x)x + (x-1)x + 3B(x-1)x + C(x-1)x + (1-x)x^2 \leq 0.$$

It suffices to have $A - 3B - C \geq 0$, which is our assumption. It is fairly straightforward to prove the last line of (A.6) is non-negative, hence we omit it here. □

Property A.2.4. Assume the following inequalities are true for constants A_{up} and B_{low}

$$\psi(A_{up}) \geq 0, \quad Ax \geq B + x + 2Bx, \quad G(B_{low}) \geq 0,$$

where

$$\begin{aligned}
\psi(A) =&(A - B - C + x - 1)\log(A - 1 - B) + (A - C)log(A) - (A - C - 1)log(A - 1) \\
&- (A - B - C + x)\log(A - B) + \log(A - B) - \log(A), \\
G(B) =&(B - x)\log(B + 1) + (A - B - C - 1 + x)log(A - 1 - B) - (B - x)\log(B) \\
&- (A - B - C + x - 1)\log(A - B).
\end{aligned}$$

For fixed B, x and C, if (A.4) is true for $A = A_{up}$, then (2.5) is true for any $A \in (\max\{B, C\}, A_{up})$; for fixed A, x and C, if (A.4) is true for $B = B_{low}$, then (2.5) is true for any $B \in (B_{low}, A)$.

Proof. Using Property A.2.1, we only need to show (A.4) is true accordingly. Same as before we use (backward) mathematical induction on A. For $A = A_{up}$ we want

$$\frac{\binom{C}{x}}{\binom{A-1}{B}} \leq \frac{\binom{C}{x}}{\binom{A}{B}}(\frac{B}{A-1})^{B-x}(\frac{A-1-B}{A-1})^{A-B-C+x-1}r^{x-B}(1-r)^{-A+B+C-x},$$

which is equivalent to $\psi(A_{up}) \geq 0$. Similarly as the proof of Property 3, it suffices to show

$$\begin{aligned}
\psi'(A) =&\frac{C+1-A}{A-1} + \frac{A-C-1}{A} + \frac{A-B-C+x-1}{A-B-1} - \frac{A-B-C+x-1}{A-B} \\
&+ \log\frac{A(A-B-1)}{(A-1)(A-B)} \leq 0,
\end{aligned}$$

for any $A \leq A_{up}$ that satisfies the assumptions. It suffices to have $B + B^2 + BC + B^2C + A(-B - 2BC - x) + A^2x \leq 0$, this only needs $B + 1 \leq A$, which is obviously true according to our assumptions. Similarly for the second part we need for $B = B_{low}$

$$\frac{\binom{C}{x}}{\binom{A}{B+1}} \leq \frac{\binom{C}{x}}{\binom{A}{B}}(\frac{B+1}{A})^{B+1-x}(\frac{A-1-B}{A})^{A-B-C+x-1}r^{x-B}(1-r)^{-A+B+C-x}.$$

It suffices to have $G(B) \geq 0$ for any $B \geq B_{low}$ that satisfies the assumptions. Again we prove the

monotonicity of $G(B)$

$$G'(B) = \frac{-B-C+A+x-1}{A-B} - \frac{-B-C+A+x-1}{-B+A-1} - \log(-B+A-1) + \log(A-B)$$
$$- \frac{B-x}{B} + \frac{B-x}{B+1} - \log(B) + \log(B+1) \geq 0.$$

It suffices to show $B + B^2 + BC + B^2C - BA - Ax - 2BAx + A^2x \geq 0$, which can be proved by using our assumption $Ax \geq B + x + 2Bx$. Thus the second part is proved. □

A.3 Lemmas of Chapter 2

To make this dissertation self-contained, in this section we list the lemmas from Short [2013] that were slightly modified to be applicable in our calculation. Detailed proof can be found in relevant references.

Lemma A.3.1. *(Bounds on Poisson distribution.) Let $U \sim Pos(m)$. Then for $p \in (0,1)$,*

$$\mathbb{P}\left[U \leq m + \Phi^{-1}(p)\sqrt{m} + \frac{\Phi^{-1}(p)^2}{6}\right] \geq p, \tag{A.7}$$

$$\mathbb{P}\left[U \geq m - \sqrt{-2m\log(1-p)}\right] \geq p, \tag{A.8}$$

here $\Phi^{-1}(\cdot)$ is the inverse cdf function of standard normal distribution.

The following inequality is tighter than Chernoff's bound.

Lemma A.3.2. *(Large deviation bound on binomial distribution.) Let $X \sim Bin(n,p)$, and $h(a,b) = a\log\frac{a}{b} + (1-a)\log\frac{1-a}{1-b}$, $a,b \in (0,1)$. Then for a fixed $t \in (0, np)$,*

$$\mathbb{P}\left[X \geq np - t\right] \geq 1 - \frac{e^{-nh(1-p+\frac{t}{n},1-p)}}{\max\{2, \sqrt{4\pi nh(1-p+\frac{t}{n},1-p)}\}}.$$

A.4 A Direct Tail Bound on Hypergeometric Distribution

Theorem A.4.1. *Let $X \sim H(A, B, C)$, then $\mathbb{E}[X] = \frac{BC}{A}$. For X and $t \geq 0$ we have*

1. *If $C \geq B$, then*

$$\mathbb{P}\left[X \geq \frac{BC}{A} - Bt\right] \geq 1 - e^{-2Bt^2}. \tag{A.9}$$

2. *If $C \leq B$, then*

$$\mathbb{P}\left[X \geq \frac{BC}{A} - Ct\right] \geq 1 - e^{-2Ct^2}. \tag{A.10}$$

Proof. The proof of (A.9) can be found in Chvátal [1979]. Assume $Q \leq B \leq C$, now we will prove (A.10) from the following three facts.

Fact 1: We have

$$\mathbb{P}[X \geq Q] = \sum_{i=Q}^{B} \binom{B}{i}\binom{A-B}{C-i}\binom{A}{C}^{-1}. \tag{A.11}$$

The equation (A.11) comes from the definition of hypergeometric distribution.

Fact 2: We have

$$\sum_{i=j}^{B} \binom{B}{i}\binom{A-B}{C-i}\binom{i}{j} = \binom{B}{j}\binom{A-j}{C-j}. \tag{A.12}$$

To prove (A.12), it suffices to note

$$\begin{aligned}
\textit{LHS of (A.12)} &= \sum_{i=j}^{B} \binom{A-B}{C-i} \frac{B!}{(B-i)!i!} \frac{i!}{(i-j)!j!} \\
&= \sum_{i=j}^{B} \binom{A-B}{C-i} \frac{B!}{j!(B-j)!} \frac{(B-j)!}{(B-i)!(i-j)!} \\
&= \binom{B}{j} \sum_{i=j}^{B} \binom{A-B}{n-i}\binom{B-j}{i-j} \\
&= \binom{B}{j}\binom{A-j}{C-j}.
\end{aligned}$$

Fact 3: We have

$$\binom{A}{C}^{-1}\binom{B}{j}\binom{A-j}{C-j} \leq \binom{B}{j}\left(\frac{C}{A}\right)^{j}. \tag{A.13}$$

To prove (A.13), it suffices to show $\binom{A}{C}^{-1}\binom{A-j}{C-j} \leq (\frac{C}{A})^{j}$, which is equivalent to

$$\frac{(C-j+1)(C-j+2)...C}{(A-j+1)(A-j+2)...A} \leq \left(\frac{C}{A}\right)^{j}.$$

Based on (A.11),(A.12) and (A.13), for $y \geq 1$ we have

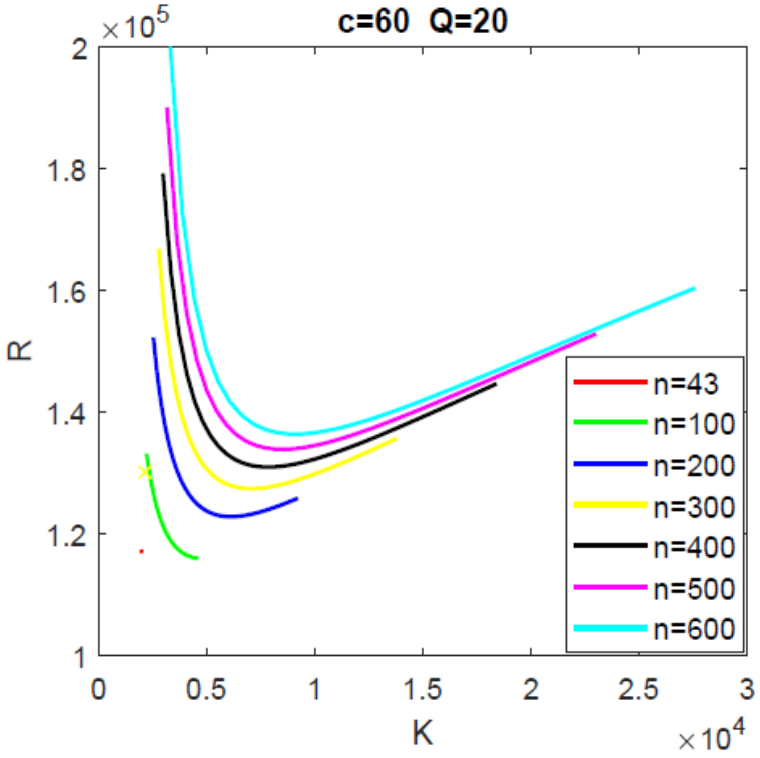
$$\begin{aligned}
\sum_{i=0}^{B}\binom{B}{i}\binom{A-B}{C-i}\binom{A}{C}^{-1}y^{i} &= \sum_{i=0}^{B}\binom{B}{i}\binom{A-B}{C-i}\binom{A}{C}^{-1}\sum_{j=0}^{i}\binom{i}{j}(y-1)^{j} && \text{(A.14)}\\
&= \sum_{j=0}^{B}\left(\binom{A}{C}^{-1}\sum_{i=j}^{B}\binom{B}{i}\binom{A-B}{C-i}\binom{i}{j}\right)(y-1)^{j}\\
&\leq \sum_{j=0}^{B}\binom{B}{j}\left(\frac{C}{A}\right)^{j}(y-1)^{j}\\
&= \left(1+\frac{(y-1)C}{A}\right)^{B}. && \text{(A.15)}
\end{aligned}$$

The LHS of (A.14) is $\mathbb{P}[X \geq y]$. The theorem follows from similar steps in Chvátal [1979]. □

A.4.1 Results from Theorem A.4.1

As a comparison, we also applied Algorithm 1 using Theorem A.4.1 as the tail bounds of Hypergeometric distribution. The result is presented in Figure A.2. Here we use the same parameters with Figure 2.3, it is fairly straightforward to see Theorem A.4.1 delivers much worse results than that of Theorem 2.3.1. However, we recommend using Theorem A.4.1 if $\frac{B}{A} \approx \frac{1}{2}$, since Theorem A.4.1 was built based on Hoeffding's inequality and Theorem 2.3.1 works better with extreme values of $\frac{B}{A}$.

Figure A.2: Results of Algorithm 1 with Theorem A.4.1



APPENDIX B

SUPPLEMENTARY MATERIALS FOR Chapter 3

B.1 Analytical Results

In this section, we introduce the detailed derivations of our optimization algorithm.

B.1.1 Formulas for Constraints

We further write $\delta_k = \phi_{k+1} - \phi_k$, $\theta_i = x_i^T \beta$. The piece-wise linear concave function $\phi(\cdot)$ can be rewritten as

$$\phi(y) = \begin{cases} \frac{\delta_k}{\Delta_k} \cdot y + \frac{\phi_k y_{(k+1)} - \phi_{k+1} y_{(k)}}{\Delta_k}, & \text{if } y \in [y_{(k)}, y_{(k+1)}], \\ -\infty, & \text{o.w.} \end{cases}$$

The identifiability constraints in (3.8) can be expanded as

$$\begin{aligned} &\sum_{k=1}^{K-1} \frac{\Delta_k (e^{\phi_{k+1}} - e^{\phi_k})}{\delta_k} = 1 \\ &\sum_{k=1}^{K-1} \left(\frac{\Delta_k}{\delta_k} \right)^2 \cdot \left\{ \left[\frac{y_{(k+1)} \delta_k}{\Delta_k} - 1 \right] \cdot e^{\phi_{k+1}} - \left[\frac{y_{(k)} \delta_k}{\Delta_k} - 1 \right] \cdot e^{\phi_k} \right\} = 1. \end{aligned}$$

B.1.2 Formulas for partial derivatives

To do the projected gradient descend, we will need the closed forms for partial derivatives. The closed form of $A(\theta_i)$ can be written as

$$A(\theta_i) = \log \sum_{k=1}^{K-1} \frac{\left[e^{\theta_i y_{k+1} + \phi_{k+1}} - e^{\theta_i y_k + \phi_k} \right] \Delta_k}{\delta_k + \theta_i \Delta_k}.$$

Let

$$J_i = \sum_{k=1}^{K-1} \frac{\left[e^{\theta_i y_{(k+1)} + \phi_{k+1}} - e^{\theta_i y_{(k)} + \phi_k} \right] \Delta_k}{\delta_k + \theta_i \Delta_k}.$$

Then we have

$$\begin{aligned} \frac{\partial J_i}{\partial \phi_k} = &\mathbb{I}_{k<K} \cdot \frac{\Delta_k e^{\theta_i y_{(k)} + \phi_k} \left(-\delta_k - \theta_i \Delta_k + e^{\theta_i \Delta_k + \delta_k} - 1 \right)}{(\delta_k + \theta_i \Delta_k)^2} \\ &+ \mathbb{I}_{k>1} \cdot \frac{\Delta_{k-1} e^{\theta_i y_{(k)} + \phi_k} \left(\delta_{k-1} + \theta_i \Delta_{k-1} - 1 + e^{-\theta_i \Delta_{k-1} - \delta_{k-1}} \right)}{(\delta_{k-1} + \theta_i \Delta_{k-1})^2}. \end{aligned}$$

We use J_{ik} to denote the above equation. Now we can calculate the partial derivative of the log-likelihood function with respect to ϕ_k as

$$\frac{\partial l_n(\phi)}{\partial \phi_k} = \frac{\partial \sum_{i=1}^{N} \phi(y_i) - \log J_i}{\partial \phi_k} = N_k - \sum_{i=1}^{N} \frac{J_{ik}}{J_i}.$$

The optimization with respect to β is relatively easier, the partial derivative is

$$\frac{\partial J_i}{\partial \beta_1} = \sum_{k=1}^{K-1} \left\{ \frac{\Delta_k \left[x_i y_{k+1} e^{\theta_i y_{k+1} + \phi_{k+1}} - x_i y_k e^{\theta_i y_k + \phi_k} \right]}{\delta_k + \theta_i \Delta_k} - \frac{\Delta_k^2 x_i \left[e^{\theta_i y_{(k+1)} + \phi_{k+1}} - e^{\theta_i y_{(k)} + \phi_k} \right]}{(\delta_k + \theta_i \Delta_k)^2} \right\}.$$

We use $J_i^{(1)}$ to denote the equation above. Similarly

$$\frac{\partial J_i}{\partial \beta_0} = \sum_{k=1}^{K-1} \left\{ \frac{\Delta_k \left[y_{(k+1)} e^{\theta_i y_{(k+1)} + \phi_{k+1}} - y_{(k)} e^{\theta_i y_{(k)} + \phi_k} \right]}{\delta_k + \theta_i \Delta_k} - \frac{\Delta_k^2 \left[e^{\theta_i y_{(k+1)} + \phi_{k+1}} - e^{\theta_i y_{(k)} + \phi_k} \right]}{(\delta_k + \theta_i \Delta_k)^2} \right\}.$$

We use $J_i^{(0)}$ to denote the equation above. To sum up

$$\frac{\partial l_n}{\partial \beta_1} = \sum_{i=1}^{N} x_i y_i - \frac{J_i^{(1)}}{J_i}, \quad \frac{\partial l_n}{\partial \beta_0} = \sum_{i=1}^{N} y_i - \frac{J_i^{(0)}}{J_i}.$$

APPENDIX C

SUPPLEMENTARY MATERIALS FOR Chapter 4

C.1 Proofs of Main Theorems in Chapter 4

In this section, we will prove the theorems from Section 4.3 under noisy case. The following Lemmas are used to prove Theorem 4.3.1.

Lemma C.1.1. *Let* $\mathbf{b}$ *be a vector sampled uniformly from* $\mathbb{S}^{d-1}$*, and* λ_k $(k = 1, .., d)$ *be constants such that* $1 \geq \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d \geq 0$*. For constant* $g_1 \in (\lambda_d, \lambda_1)$*, we write* $r_i = (g_1^2 - \lambda_i^2)_+$ *and* $s_i = (g_1^2 - \lambda_i^2)_-$*. Assuming that* $\sum_{i=1}^d r_i > \sum_{i=1}^d s_i$*, then*

$$\mathbb{P}\left[\sum_{i=1}^{d} |\lambda_i b_i|^2 < g_1^2\right] \geq 1 - 2e^{-\epsilon^2},$$

where

$$\epsilon = \frac{\sum_{i=1}^d (r_i - s_i)}{(\sqrt{\sum_{i=1}^d r_i^2} + \sqrt{\sum_{i=1}^d s_i^2}) + \sqrt{(\sqrt{\sum_{i=1}^d r_i^2} + \sqrt{\sum_{i=1}^d s_i^2})^2 + 2s_1 \sum_{i=1}^d (r_i - s_i)}}.$$

Proof. We write $b_i = \frac{z_i}{\sqrt{\sum_{j=1}^d z_j^2}}$, where $\{z_i\}_{i=1}^d$ are i.i.d. $\mathcal{N}(0,1)$ random variables. The goal is to bound

$$\mathbb{P}\left[\sum_{i=1}^{d} (g_1^2 - \lambda_i^2)_- \cdot z_i^2 \geq \sum_{i=1}^{d} (g_1^2 - \lambda_i^2)_+ \cdot z_i^2\right] = \mathbb{P}\left[\sum_{i=1}^{d} s_i \cdot z_i^2 \geq \sum_{i=1}^{d} r_i \cdot z_i^2\right].$$

Note that $g_1 \in (\lambda_d, \lambda_1)$, hence both $\sum_{i=1}^d r_i$ and $\sum_{i=1}^d s_i$ are strictly positive.

Now we write $X = \sum_{i=1}^d s_i \cdot z_i^2$ and $Y = \sum_{i=1}^d r_i \cdot z_i^2$. Applying Lemma 1 in Laurent and Massart [2000] we have for positive constants ϵ_1 and ϵ_2

$$\mathbb{P}\left[X \geq \sum_{i=1}^{d} s_i + 2\sqrt{\sum_{i=1}^{d} s_i^2}\epsilon_1 + 2s_1\epsilon_1^2\right] \leq e^{-\epsilon_1^2}, \quad \mathbb{P}\left[Y \leq \sum_{i=1}^{d} r_i - 2\sqrt{\sum_{i=1}^{d} r_i^2}\epsilon_2\right] \leq e^{-\epsilon_2^2}.$$

We set $\epsilon_1 = \epsilon_2$ and

$$\sum_{i=1}^{d} s_i + 2\sqrt{\sum_{i=1}^{d} s_i^2}\epsilon_1 + 2s_1\epsilon_1^2 = \sum_{i=1}^{d} r_i - 2\sqrt{\sum_{i=1}^{d} r_i^2}\epsilon_2.$$

Solving the above quadratic equation we have

$$\epsilon_1 = \epsilon_2 = \frac{\sum_{i=1}^{d}(r_i - s_i)}{(\sqrt{\sum_{i=1}^{d} r_i^2} + \sqrt{\sum_{i=1}^{d} s_i^2}) + \sqrt{(\sqrt{\sum_{i=1}^{d} r_i^2} + \sqrt{\sum_{i=1}^{d} s_i^2})^2 + 2s_1\sum_{i=1}^{d}(r_i - s_i)}}.$$

Consequently

$$\begin{aligned}\mathbb{P}[X \geq Y] &\leq \mathbb{P}[X \geq \sum_{i=1}^{d} s_i + 2\sqrt{\sum_{i=1}^{d} s_i^2}\epsilon_1 + 2s_1\epsilon_1^2] + \mathbb{P}[Y \leq \sum_{i=1}^{d} r_i - 2\sqrt{\sum_{i=1}^{d} r_i^2}\epsilon_2] \\ &\leq e^{-\epsilon_1^2} + e^{-\epsilon_2^2}.\end{aligned}$$

Substituting ϵ_1 and ϵ_2 into the inequality above yields the result. □

The following bound on F-distributed random variables follows from Lemma C.1.1.

Corollary C.1.1. *Let $X \sim F(m, n)$, and $m, n \geq 2$. Then for constant $q > 1$, we have*

$$\mathbb{P}\left[X \geq q\right] \leq 2e^{-\epsilon^2},$$

where $\epsilon = \frac{1}{2}[-(\sqrt{m} + \frac{qm}{\sqrt{n}}) + \sqrt{(\sqrt{m} + \frac{qm}{\sqrt{n}})^2 + 2m(q-1)}]$.

Proof. We write $b_i = \frac{z_i}{\sum_{i=1}^{m+n} z_i^2}$, and $X = \frac{(\sum_{i=1}^{m} z_i^2)/m}{(\sum_{i=m+1}^{m+n} z_i^2)/n}$, where $\{z_i\}_{i=1}^{m+n}$ are i.i.d. $\mathcal{N}(0,1)$ random variables. It follows

$$\mathbb{P}\left[X \geq q\right] = \mathbb{P}\left[\sum_{i=1}^{m} \frac{1}{mq} \cdot z_i^2 \geq \sum_{i=m+1}^{m+n} \frac{1}{n} \cdot z_i^2\right].$$

The corollary follows by selecting $\lambda_i^2 = \frac{1}{2} + \frac{1}{mq}$ for $i = 1, ..., m$, $\lambda_i^2 = \frac{1}{2} - \frac{1}{n}$ for $i = m+1, ..., m+n$, and $g_1^2 = \frac{1}{2}$ in Lemma C.1.1. □

Lemma C.1.2 states a bound on the order statistics of Beta distributed random variables.

Lemma C.1.2. *Assume T_u and T_l satisfy Assumption A1. For any $k = 1, ..., K$, let $\{B_{(i)}\}_{i=1}^{N_k-1}$ be the order statistics from a sample of $(N_k - 1)$ i.i.d $\beta(\frac{1}{2}, \frac{d-1}{2})$ random variables, then*

$$\max\left(\mathbb{P}\left[B_{(N_k-d_{\max})} \leq T_l^2\right], \mathbb{P}\left[B_{(N_k-d_{\max})} \geq T_u^2\right]\right) \leq \frac{(N_k - d_{\max})}{d_{\max}(N_k+1)(N_k^{\rho}-1)^2}.$$

Proof. Let $U_{(i)} = F_{(\frac{1}{2},\frac{d-1}{2})}(B_{(i)})$, here $F_{(\frac{1}{2},\frac{d-1}{2})}$ is the CDF of the Beta distribution $\beta(\frac{1}{2}, \frac{d-1}{2})$. Note that $\{U_{(i)}\}_{i=1}^{N_k-1}$ are the order statistics of the uniform distribution.

From Assumption $A1$ we know $F_{(\frac{1}{2},\frac{d-1}{2})}(T_l^2) \leq 1 - \frac{d_{\max}}{N_k^{1-\rho}}$ and hence

$$\mathbb{P}\left[B_{(N_k-d_{\max})} \leq T_l^2\right] \leq \mathbb{P}\left[U_{(N_k-d_{\max})} \leq 1 - \frac{d_{\max}}{N_k^{1-\rho}}\right]. \tag{C.1}$$

By Chebyshev's inequality and basic properties of the uniform order statistics we have

$$\mathbb{P}\left[U_{(N_k-d_{\max})} \leq 1 - \frac{d_{\max}}{N_k^{1-\rho}}\right] \leq \frac{Var\left[U_{(N_k-d_{\max})}\right]}{(\frac{d_{\max}}{N_k} - \frac{d_{\max}}{N_k^{1-\rho}})^2} = \frac{(N_k - d_{\max})}{d_{\max}(N_k+1)(N_k^{\rho}-1)^2}. \tag{C.2}$$

Combine (C.1) and (C.2) we know

$$\mathbb{P}\left[B_{(N_k-d_{\max})} \leq T_l^2\right] \leq \frac{(N_k - d_{\max})}{d_{\max}(N_k+1)(N_k^{\rho}-1)^2}.$$

Similarly we can prove

$$\mathbb{P}\left[B_{(N_k-d_{\max})} \geq T^2\right] \leq \frac{(N_k - d_{\max})}{d_{\max}(N_k+1)(N_k^{\rho}-1)^2}.$$

This completes the proof. □

Lemma C.1.3 to Lemma C.1.5 are used to prove Theorem 4.3.2.

Lemma C.1.3. *Let $\mathbf{v}$ be a random vector that uniformly distributed on $\mathbb{S}^{d-1}$. Then we can decompose $\mathbf{v}$ into $\mathbf{v} = \left[\sqrt{g}s, \sqrt{1-g}\mathbf{u}\right]$, where $g \sim \beta(\frac{1}{2}, \frac{d-1}{2})$, $\mathbf{u} \sim U(\mathbb{S}^{d-2})$ and $\mathbb{P}[s = 1] = \mathbb{P}[s = -1] = 0.5$ are three independent random variables.*

Proof. It is straightforward to see $\langle \mathbf{v}, \mathbf{v} \rangle = [v_1^2, ..., v_d^2]$ follows the Dirichlet distribution with parameters $\boldsymbol{\alpha} = (\frac{1}{2}, ..., \frac{1}{2}) \in \mathbb{R}^d$. We can decompose $\langle \mathbf{v}, \mathbf{v} \rangle$ into the following concatenation of two random

components

$$[v_1^2, ..., v_d^2] = [v_1^2, (1 - v_1^2)\frac{\langle \mathbf{v}_{-1}, \mathbf{v}_{-1} \rangle}{1 - v_1^2}].$$

Since Dirichlet distribution is completely neutral [Lin, 2016], we know that v_1^2 is independent of $\frac{\langle \mathbf{v}_{-1}, \mathbf{v}_{-1} \rangle}{1-v_1^2}$, where $v_1^2 \sim \beta(\frac{1}{2}, \frac{d-1}{2})$ and $\frac{\langle \mathbf{v}_{-1}, \mathbf{v}_{-1} \rangle}{1-v_1^2} \sim Dir(\boldsymbol{\alpha}_{-1})$. From symmetry, we can let $v_1 = \sqrt{g}s$ and $\frac{\mathbf{v}_{-1}}{\sqrt{1-v_1^2}} = \mathbf{u}$, where the distributions of g, $\mathbf{u}$ and s are specified in the statement of Lemma C.1.3. This completes the proof.

□

Let $\{\mathbf{a}_i\}_{i=1}^{N_k-1}$ be $N_k - 1$ vectors that are uniformly sampled from $\mathbb{S}^{d-1}$. From Lemma C.1.3, we know that for any $i = 1, ..., N_k - 1$, the value of a_{i1} is independent of $\frac{[a_{i2}, .., a_{id}]}{\sqrt{1-a_{i1}^2}}$. The following corollary is then a direct result from this fact.

Corollary C.1.2. *Let $\{\mathbf{a}_{(i)}\}_{i=1}^{N_k-1}$ be a permutation of $\{\mathbf{a}_i\}_{i=1}^{N_k-1}$ sorted in ascending order of the absolute value of the first coordinate. Then we can write*

$$\mathbf{a}_{(i)} = [a_{(i)1}, \sqrt{1 - a_{(i)1}^2}\mathbf{b}_{N_k-i}],$$

where $\{\mathbf{b}_i\}_{i=1}^{N_k-1}$ are i.i.d. uniform samples on $\mathbb{S}^{d-2}$.

Lemma C.1.4. *[Lerman et al., 2012, Lemma B.3] Let $\{\mathbf{b}_i\}_{i=1}^{d_{max}}$ be i.i.d. uniform samples from $\mathbb{S}^{d-2}$, $d \geq 3$. Then for any $t \geq 0$:*

$$\inf_{\|\mathbf{u}\|_2=1} \sum_{i=1}^{d_{max}} |\langle \mathbf{u}, \mathbf{b}_i \rangle| \geq \sqrt{\frac{2}{\pi}}\frac{d_{max}}{\sqrt{d}} - 2\sqrt{d_{max}} - t\sqrt{\frac{d_{max}}{d-1}},$$

with probability at least $1 - e^{-t^2/2}$.

Corollary C.1.3. *Use the same definition of $\{\mathbf{b}_i\}_{i=1}^{d_{max}}$ from Lemma C.1.4. Then for any $t \geq 0$*

$$\sup_{\|\mathbf{u}\|_2=1} \sum_{i=1}^{d_{max}} \langle \mathbf{u}, \mathbf{b}_i \rangle \leq 2\sqrt{d_{max}} + t\sqrt{\frac{d_{max}}{d-1}},$$

with probability at least $1 - e^{-t^2/2}$.

Proof. Note that $\mathbb{E}[\langle \mathbf{u}, \mathbf{b} \rangle] = 0$ for any $\mathbf{b} \sim U(\mathbb{S}^{d-1})$ and $\mathbf{u} \in \mathbb{R}^d$. Therefore by Lemma 6.3 in Ledoux

and Talagrand [2013] we have:

$$\mathbb{E}\left[\sup_{\|\mathbf{u}\|_2=1}\sum_{i=1}^{d_{max}}\langle \mathbf{u}, \mathbf{b}_i\rangle\right] \leq 2\sup_{\|\mathbf{u}\|_2=1}\left[\mathbb{E}\left\|\sum_{i=1}^{d_{max}}\epsilon_i \mathbf{b}_i\right\|^2\right] = 2\sqrt{d_{max}}.$$

Here $\{\epsilon_i\}_{i=1}^{d_{max}}$ are i.i.d. Rademacher random variables. The lemma is proved by following similar steps after equation (B.11) in Lerman et al. [2012]. □

Lemma C.1.5. *Suppose Assumption A3. Write* $\mathbf{a}_0 = [1, 0, ..., 0] \in \mathbb{R}^d$*, and use the definitions for* $\{\mathbf{a}_i\}_{i=1}^{N_k-1}$ *and* $\{\mathbf{a}_{(i)}\}_{i=1}^{N_k-1}$ *from Corollary C.1.2. Let* $\mathbf{B} \in \mathbb{R}^{d\times(d_{max}+1)}$ *be a matrix where its first column is* $\mathbf{a}_0$ *and its i-th column* $(2 \leq i \leq d_{max}+1)$ *is* $\mathbf{a}_{(N_k-i+1)}$*. Let the largest d singular values of* $\mathbf{B}$ *be* $s_1 \geq s_2 \geq \cdots \geq s_d$*. Then we have*

$$\mathbb{P}\left[s_d^2 \geq q_0\right] \geq 1 - \frac{2}{N^{t^2/2}} - \frac{2(N_k - d_{\max})}{d_{\max}(N_k+1)(N_k^\rho - 1)^2}.$$

Proof. From Corollary C.1.2, we know $\mathbf{B}$ can be re-written as

$$\mathbf{B} = \begin{pmatrix} 1, & a_{(N_k-1)1}, & ... & a_{(N_k-d_{max})1} \\ \mathbf{0}, & \sqrt{1-a^2_{(N_k-1)1}}\mathbf{b}_1, & ... & \sqrt{1-a^2_{(N_k-d_{max})1}}\mathbf{b}_{d_{max}} \end{pmatrix},$$

where $\{\mathbf{b}_i\}_{i=1}^{d_{max}}$ are i.i.d. uniform samples from $\mathbb{S}^{d-2}$.

Given the dimensions of $\mathbf{B}$, we know $s_d = \inf_{\|\mathbf{x}\|_2=1}\|\mathbf{B}^T\mathbf{x}\|_2$. For convenience, we write

$$\mathbf{x}' = \frac{1}{\sqrt{1-x_1^2}}\left[x_2, ..., x_d\right],$$

where $\|\mathbf{x}'\|_2 = 1$, $c_i = \langle \mathbf{x}', \mathbf{b}_i\rangle$, $a_{(N_k)1} = 1$. Let $\mathcal{E}_1$ be the event that $\{s_d^2 \geq q_0\}$, and $\mathcal{E}_2$ be the event that $\{a^2_{(N_k-i)1} \in \left[T_l^2, T_u^2\right], \forall i = 1, .., d_{max}\}$. From our assumptions, Lemma C.1.2 and Lemma C.1.5, we know

$$\mathbb{P}\left[\mathcal{E}_2\right] \geq 1 - \frac{2(N_k - d_{\max})}{d_{\max}(N_k+1)(N_k^\rho - 1)^2}. \tag{C.3}$$

Conditioning on $\mathcal{E}_2$, we have:

$$\begin{aligned}
\|\mathbf{B}^T\mathbf{x}\|_2^2 =& \left\| \begin{pmatrix} 1, & a_{(N_k-1)1}, & \dots & a_{(N_k-d_{max})1} \\ \mathbf{0}, & \sqrt{1-a_{(N_k-1)1}^2}\mathbf{b}_1, & \dots & \sqrt{1-a_{(N_k-d_{max})1}^2}\mathbf{b}_{d_{max}} \end{pmatrix}^T \mathbf{x} \right\|_2^2 \\
=& \sum_{i=0}^{d_{max}} \left(a_{(N_k-i)1}x_1 + \sqrt{\left(1-a_{(N-i)1}^2\right)\left(1-x_1^2\right)}c_i \right)^2 \\
=& \sum_{i=0}^{d_{max}} a_{(N-i)1}^2 x_1^2 + 2\sum_{i=0}^{d_{max}} \sqrt{a_{(N-i)1}^2(1-a_{(N_k-i)1}^2)(1-x_1^2)}c_i x_1 \\
&+ \sum_{i=1}^{d_{max}} (1-a_{(N_k-i)1}^2)(1-x_1^2)c_i^2 \\
\geq& T_l^2 d_{max}\cdot x_1^2 - 2\sqrt{T_u^2(1-T_u^2)(1-x_1^2)x_1^2} \sup_{\|u\|_2=1} \sum_{i=1}^{d_{max}} \langle \mathbf{u}, \mathbf{b}_i\rangle \\
&+ (1-x_1^2)(1-T_u^2) \inf_{\|u\|_2=1} \sum_{i=1}^{d_{max}} \langle \mathbf{u}, \mathbf{b}_i\rangle^2. \qquad \text{(C.4)}
\end{aligned}$$

From Lemma C.1.4 and Corollary C.1.3, we know that conditional on $\mathcal{E}_2$

$$\text{(C.4)} \geq (1-x_1^2)C_2 - 2\sqrt{T_u^2(1-T_u^2)(1-x_1^2)x_1^2}C_1 + T_l^2 d_{max}\cdot x_1^2, \qquad \text{(C.5)}$$

with probability at least that $1-\frac{2}{N^{t^2/2}}$. Since $1-x_1^2 \leq 1$, a lower bound of the RHS of (C.5) is

$$(T_l^2 d_{max} - C_2)x_1^2 - 2\sqrt{T_u^2(1-T_u^2)}C_1 x_1 + C_2 \geq \frac{(T_l^2 d_{max}-C_2)C_2 - T_u^2(1-T_u^2)C_1^2}{T_l^2 d_{max}} \geq q_0,$$

where the q_0 comes from Assumption A3. Finally, note the following fact

$$\begin{aligned}
\mathbb{P}[\mathcal{E}_1] &\geq \mathbb{P}[\mathcal{E}_1|\mathcal{E}_2] + \mathbb{P}[\mathcal{E}_2] - 1 \qquad \text{(C.6)} \\
&= 1 - \frac{2}{N^{t^2/2}} - \frac{2(N_k - d_{\max})}{d_{\max}(N_k+1)(N_k^\rho - 1)^2}.
\end{aligned}$$

This completes the proof.

□

Proof of Theorem 4.3.1. Let the event $\mathcal{E}_{1i} = \{\mathbf{Y}_{\mathcal{C}_i}$ only contains points in same subspace$\}$, then

$\mathcal{E}_1 = \cap_{i=1}^n \mathcal{E}_{1i}$ is the event that Algorithm 3 has sub-cluster preserving property. Let the event $\mathcal{E}_2 = \{\sigma||\mathbf{e}_i^{(k)}||_2 < g_2, \forall i, k\}$, where g_2 is from assumption $A2$. Our goal is to find a lower bound on $\mathbb{P}[\mathcal{E}_1]$.

Note the following fact

$$\mathbb{P}[\mathcal{E}_1] \geq 1 - \sum_{i=1}^n \mathbb{P}[\mathcal{E}_{1i}^{\complement}|\mathcal{E}_2] + \mathbb{P}[\mathcal{E}_2] - 1 = \mathbb{P}[\mathcal{E}_2] - \sum_{i=1}^n \mathbb{P}[\mathcal{E}_{1i}^{\complement}|\mathcal{E}_2]. \tag{C.7}$$

Therefore, it suffices to find a lower bound on $\mathbb{P}[\mathcal{E}_2] - \sum_{i=1}^n \mathbb{P}[\mathcal{E}_{1i}^{\complement}|\mathcal{E}_2]$.

We start by finding a preliminary upper bound on $\mathbb{P}[\mathcal{E}_{11}^{\complement}|\mathcal{E}_2]$. WLOG assume that $\mathbf{y_1}^{(1)}$ is one of the sampled points, and $\mathbf{Y}_{\mathcal{C}_1}$ is the sub-cluster associated with it. Write $\hat{A}_i^k = |\langle \mathbf{y}_1^{(1)}, \mathbf{y}_i^{(k)} \rangle|$. For $\mathcal{E}_{11}$ to happen, we need the largest $(d_{\max}+1)$ values among $\cup_{k=1}^K \hat{A}_i^k$ to be from the same set $\hat{A}_i^1$. Mathematically this means

$$\mathcal{E}_{11}^{\complement} = \left\{ \hat{A}^1_{(N_1 - d_{\max})} \leq \max_{k \neq 1} \max_{i=1,..,N_k} \hat{A}_i^k \right\}.$$

Recall from (4.1) that $\mathbf{y_i}^{(k)} = \frac{\mathbf{U_k a}_i^{(k)} + \sigma \mathbf{e}_i^{(k)}}{||\mathbf{U}_k \mathbf{a}_i^{(k)} + \sigma \mathbf{e}_i^{(k)}||_2}$. The triangle inequality tells us that:

$$||\mathbf{U}_k \mathbf{a}_i^{(k)}||_2 - ||\sigma \mathbf{e}_i^{(k)}||_2 \leq ||\mathbf{U}_k \mathbf{a}_i^{(k)} + \sigma \mathbf{e}_i^{(k)}||_2 \leq ||\mathbf{U}_k \mathbf{a}_i^{(k)}||_2 + ||\sigma \mathbf{e}||_2.$$

Therefore conditional on $\mathcal{E}_2$, we know the normalizing constants $||\mathbf{U}_k \mathbf{a}_i^{(k)} + \sigma \mathbf{e}_i^{(k)}||_2$ are bounded in $[1 - g_2, 1 + g_2]$. For fixed $\mathbf{y}_1^{(1)}$, we write $A_i^k = ||\mathbf{y}_1^{(1)}||_2 \cdot ||\mathbf{y}_i^{(k)}||_2 \cdot \hat{A}_i^k$. It is fairly straightforward to get the following relation

$$\mathbb{P}\left[A^1_{(N_1 - d_{\max})} \leq \frac{1+g_2}{1-g_2} \max_{k \neq 1} \max_{1 \leq i \leq N_k} A_i^k \middle| \mathcal{E}_2 \right] \geq \mathbb{P}\left[\mathcal{E}_{11}^{\complement} \middle| \mathcal{E}_2 \right]. \tag{C.8}$$

Conditioning on $\mathcal{E}_2$ and write $B_i = \left| \langle \mathbf{a}_1^{(1)}, \mathbf{a}_i^{(1)} \rangle \right|^2$, $i = 2, ..., N_1 - 1$. We have the following

inequalities

$$\begin{aligned}
A^1_{(N_1-d_{\max})} =& \left|\sqrt{B_{(N_1-d_{\max})}} + \sigma\langle \mathbf{U}_1\mathbf{a}_1^{(1)}, \mathbf{e}_i^{(1)}\rangle + \sigma\langle \mathbf{U}_1\mathbf{a}_i^{(1)}, \mathbf{e}_1^{(1)}\rangle + \sigma^2\langle \mathbf{e}_1^{(1)}, \mathbf{e}_i^{(1)}\rangle\right| \\
\geq& \sqrt{B_{(N_1-d_{\max})}} - \sigma||\mathbf{e}_i^{(1)}||_2 - \sigma||\mathbf{e}_1^{(1)}||_2 - \sigma^2||\mathbf{e}_1^{(1)}||_2 \max_{i\neq 1}||\mathbf{e}_i^{(1)}||_2 \\
\geq& \sqrt{B_{(N_1-d_{\max})}} - 2g_2 - g_2^2.
\end{aligned}$$

Similarly we have

$$\begin{aligned}
\max_{k\neq 1}\max_{1\leq i\leq N_k} A_i^k =& \max_{k\neq 1}\max_{1\leq i\leq N_k}\left|\langle \mathbf{U}_1\mathbf{a}_1^{(1)}, \mathbf{U}_k\mathbf{a}_i^{(k)}\rangle + \sigma\langle \mathbf{U}_1\mathbf{a}_1^{(1)}, \mathbf{e}_i^{(k)}\rangle + \sigma\langle \mathbf{U}_k\mathbf{a}_i^{(k)}, \mathbf{e}_1^{(1)}\rangle + \sigma^2\langle \mathbf{e}_1^{(1)}, \mathbf{e}_i^{(k)}\rangle\right| \\
\leq& \max_{k\neq 1}\max_{1\leq i\leq N_k}\left|\langle \mathbf{U}_1\mathbf{a}_1^{(1)}, \mathbf{U}_k\mathbf{a}_i^{(k)}\rangle\right| + \sigma\max_{k\neq 1}\max_{1\leq i\leq N_k}||\mathbf{e}_i^{(k)}||_2 \\
&+ \sigma||\mathbf{e}_1^{(1)}||_2 + \sigma^2||\mathbf{e}_1^{(1)}||_2\max_{k\neq 1}\max_{1\leq i\leq N_k}||\mathbf{e}_i^{(k)}||_2 \\
\leq& \max_{k\neq 1}\max_{1\leq i\leq N_k}\left|\langle \mathbf{U}_1\mathbf{a}_1^{(1)}, \mathbf{U}_k\mathbf{a}_i^{(k)}\rangle\right| + 2g_2 + g_2^2.
\end{aligned}$$

Pick T from assumption $A3$, then the LHS of (C.8) has the following upper bound

$$\mathbb{P}\left[T \leq Q|\mathcal{E}_2\right] + \mathbb{P}\left[B_{(N_1-d_{\max})} \leq T^2\middle|\mathcal{E}_2\right], \tag{C.9}$$

where

$$Q = (1+\frac{1+g_2}{1-g_2})(2g_2+g_2^2) + \frac{1+g_2}{1-g_2}\max_{k\neq 1}\max_{1\leq i\leq N_k}\left|\langle \mathbf{U}_1\mathbf{a}_1^{(1)}, \mathbf{U}_k\mathbf{a}_i^{(k)}\rangle\right|.$$

Now we are going to complete our proof in 3 steps.

Step 1: For the first term in (C.9) we have

$$\mathbb{P}\left[T \leq Q|\mathcal{E}_2\right] = \mathbb{P}\left[g_1 \leq \max_{k\neq 1}\max_{1\leq i\leq N_k}\left|\langle \mathbf{U}_1\mathbf{a}_1^{(1)}, \mathbf{U}_k\mathbf{a}_i^{(k)}\rangle\right|\right].$$

From singular value decomposition we can write

$$\langle \mathbf{U}_1\mathbf{a}_1^{(1)}, \mathbf{U}_k\mathbf{a}_i^{(k)}\rangle = \mathbf{a}_1^{(1)T}\mathbf{W}_{1k}\mathbf{\Lambda}_{1k}\mathbf{V}_{1k}^T\mathbf{a}_i^{(k)} := \mathbf{b}_k^T\mathbf{\Lambda}_{1k}\mathbf{V}_{1k}^T\mathbf{a_i}^{(k)},$$

where both $\{\mathbf{b}_k\}_{k=2}^K$ and $\{\mathbf{V}_{1k}^T\mathbf{a}_i^{(k)}\}_{k=2}^K$ are sampled uniformly from $\mathbb{S}^{d-1}$. Therefore

$$\begin{aligned}
\mathbb{P}\left[g_1 \le \max_{k\neq 1}\max_{1\le i\le N_k}\left|\langle \mathbf{U}_1\mathbf{a}_1^{(1)}, \mathbf{U}_k\mathbf{a}_i^{(k)}\rangle\right|\right] =& \mathbb{P}\left[g_1^2 \le \max_{k\neq 1}\max_{1\le i\le N_k}\left|\mathbf{b}_k^T\mathbf{\Lambda}_{1k}\mathbf{V}_{1k}^T\mathbf{a}_i^{(k)}\right|^2\right] \\
\le& \sum_{k=2}^K \mathbb{P}\left[g_1^2 \le \max_{1\le i\le N_k}\left|\mathbf{b}_k^T\mathbf{\Lambda}_{1k}\mathbf{V}_{1k}^T\mathbf{a}_i^{(k)}\right|^2\right] \qquad \text{(C.10)} \\
\le& \sum_{k=2}^K \mathbb{P}\left[g_1^2 \le \sum_{i=1}^d\left|\lambda_i^{(1k)}b_{ki}\right|^2\right] \qquad \text{(C.11)} \\
\le& \sum_{k=2}^K \mathbb{P}\left[g_1^2 \le \sum_{i=1}^d\left|\lambda_i^{(1)}b_{ki}\right|^2\right], \qquad \text{(C.12)}
\end{aligned}$$

where inequality (C.10) uses the union bound inequality, (C.11) comes from Cauchy-Schwarz inequality, and (C.12) uses Definition 4.3.1. Since $\{\mathbf{b}_k\}_{k=2}^K \sim U(\mathbb{S}^{d-1})$, we can write (C.12) as

$$(K-1)\mathbb{P}\left[g_1^2 \le \sum_{i=1}^d(\lambda_i^{(1)}b_i)^2\right],$$

where $\mathbf{b}$ is uniformly distributed on $\mathbb{S}^{d-1}$. Now we apply Lemma C.1.1 directly to the quantity above and get $\mathbb{P}\left[g_1^2 \le \sum_{i=1}^d\left(\lambda_i^{(1)}b_i\right)^2\right] \le 2e^{-\epsilon'^2}$ where[1]

$$\begin{aligned}
\epsilon' &= \frac{\sum_{i=1}^d(r_i-s_i)}{(\sqrt{\sum_{i=1}^d r_i^2}+\sqrt{\sum_{i=1}^d s_i^2})+\sqrt{(\sqrt{\sum_{i=1}^d r_i^2}+\sqrt{\sum_{i=1}^d s_i^2})^2+2s_1\sum_{i=1}^d(r_i-s_i)}} \\
&= \frac{-(\sqrt{\sum_{i=1}^d r_i^2}+\sqrt{\sum_{i=1}^d s_i^2})+\sqrt{(\sqrt{\sum_{i=1}^d r_i^2}+\sqrt{\sum_{i=1}^d s_i^2})^2+2s_1\sum_{i=1}^d(r_i-s_i)}}{2s_1} \\
&\ge \frac{-(\sqrt{\sum_{i=1}^d r_i^2}+\sqrt{\sum_{i=1}^d s_i^2})+\sqrt{(\sqrt{\sum_{i=1}^d r_i^2}+\sqrt{\sum_{i=1}^d s_i^2})^2+2\sum_{i=1}^d(r_i-s_i)}}{2} \qquad \text{(C.13)} \\
&= \frac{\sum_{i=1}^d(r_i-s_i)}{(\sqrt{\sum_{i=1}^d r_i^2}+\sqrt{\sum_{i=1}^d s_i^2})+\sqrt{(\sqrt{\sum_{i=1}^d r_i^2}+\sqrt{\sum_{i=1}^d s_i^2})^2+2\sum_{i=1}^d(r_i-s_i)}} \\
&\ge \frac{\sum_{i=1}^d(r_i-s_i)}{2\sqrt{\sum_{i=1}^d r_i^2}+\sqrt{4\sum_{i=1}^d r_i^2+2\sum_{i=1}^d r_i}} \ge \epsilon.
\end{aligned}$$

[1] r_i and s_i are defined in Lemma C.1.1.

Here ϵ is defined in Theorem 4.3.1 and (C.13) comes from the following fact for positive constants a, b and $s \in (0,1)$

$$\frac{-a+\sqrt{a^2+2sb}}{2s} \geq \frac{-a+\sqrt{a^2+2b}}{2}.$$

Therefore we have

$$\mathbb{P}\left[g_1 \leq \max_{k\neq 1}\max_{1\leq i\leq n_k}\left|\langle \mathbf{U}_1\mathbf{a}_1^{(1)}, \mathbf{U}_k\mathbf{a}_i^{(k)}\rangle\right|\right] \leq 2(K-1)e^{-\epsilon^2}.$$

Step 2: For the second term of (C.9), we just need to use Lemma C.1.2. Note that for fixed $\mathbf{a}_1^{(1)}$, one can show $B_i = \langle \mathbf{a}_1^{(1)}, \mathbf{a}_i^{(1)}\rangle^2$ can be treated as a sample from a Beta distribution with parameters $(\frac{1}{2}, \frac{d-1}{2})$. From Lemma C.1.2 and Assumption $A3$ we have

$$\mathbb{P}\left[B_{(N_1-d_{\max})} \leq T^2 \middle| \mathcal{E}_2\right] \leq \frac{(N_1-d_{\max})}{d_{\max}(N_1+1)(N_1^{\rho}-1)^2}.$$

Combine the results above we know

$$\mathbb{P}\left[\mathcal{E}_{11}^{\complement}\middle|\mathcal{E}_2\right] \leq 2(K-1)e^{-\epsilon^2} + \frac{(N_1-d_{\max})}{d_{\max}(N_1+1)(N_1^{\rho}-1)^2}. \tag{C.14}$$

Step 3: Now we are going to find the lower bound on $\mathbb{P}[\mathcal{E}_2]$. Let $\mathbf{e}$ be an independent copy of $\mathbf{e}_1^{(1)}$, note that $\frac{||\mathbf{e}||_2^2}{D} \sim F_{D,d}$. From Lemma C.1.5 we have

$$\mathbb{P}\left[g_2 \leq \sigma||\mathbf{e}||_2\right] = \mathbb{P}\left[\frac{g_2^2}{D\sigma^2} \leq \frac{||\mathbf{e}||_2^2}{D}\right] \leq 2e^{-t^2},$$

where t can be found as Corollary C.1.1. Using Assumption $A2$ we have

$$t > \frac{D(\frac{g_2^2}{D\sigma^2}-1)}{2(\sqrt{D}+\frac{g_2^2}{\sigma^2\sqrt{d}}+\sqrt{d})} = \frac{\sqrt{d}}{2}(1-\frac{1+\frac{d}{D}+\sqrt{\frac{d}{D}}}{1+\frac{d}{D}+\frac{g_2^2}{D\sigma^2}}) \geq (1+\frac{\eta}{2+\eta})\sqrt{\log N}.$$

Therefore we have $\mathbb{P}\left[g_2 \leq \sigma||\mathbf{e}||_2\right] \leq \frac{2}{N^{(1+\frac{\eta}{2+\eta})^2}}$. Now we note that

$$\begin{aligned}\mathbb{P}\left[g_2 \leq \sigma \max_{k=1,...,K} \max_{1\leq i\leq N_k} ||\mathbf{e}_i^{(k)}||_2\right] &= 1 - \mathbb{P}\left[g_2 > \sigma \max_{k=1,...,K} \max_{1\leq i\leq N_k} ||\mathbf{e}_i^{(k)}||_2\right] \\ &= 1 - \Pi_{i=1}^{N}(1 - \mathbb{P}\left[g_2 \leq \sigma||\mathbf{e}_i^{(k)}||_2\right]) \\ &\leq 1 - (1 - 2e^{-t^2})^N \leq \frac{2N}{N^{(1+\frac{\eta}{2+\eta})^2} - 2},\end{aligned}$$

where the last inequality comes from the Taylor expansion of $\exp\left(\frac{-2N}{N^{(1+\frac{\eta}{2+\eta})^2}-2}\right)$. Therefore

$$\mathbb{P}\left[\mathcal{E}_2\right] \geq 1 - \frac{2N}{N^{(1+\frac{\eta}{2+\eta})^2} - 2}. \tag{C.15}$$

Finally, the above arguments hold for any $\mathbf{y}_i^{(k)}$. Putting (C.14) and (C.15) together and applying the union bound inequality yields the result

$$\mathbb{P}[\mathcal{E}_1] \geq 1 - \sum_{k=1}^{K} \frac{n_k(N_k - d_{\max})}{d_{\max}(N_k+1)(N_k^{\rho}-1)^2} - 2(K-1)ne^{-\epsilon^2} - \frac{2N}{N^{(1+\frac{\eta}{2+\eta})^2} - 2}. \tag{C.16}$$

□

To prove Theorem 4.3.2, we will use the following equation

$$(\mathbf{W}^T\mathbf{W} + \lambda\mathbf{I}_{d_2})^{-1}\mathbf{W}^T = \mathbf{W}^T(\mathbf{W}\mathbf{W}^T + \lambda\mathbf{I}_{d_1})^{-1}, \tag{C.17}$$

where $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$ and λ is a positive constant [Murphy, 2012, Chapter 4]. Throughout the proof of Theorem 4.3.2, the subscript of identity matrix $\mathbf{I}$ will be omitted as its dimension is clear from the context.

Proof of Theorem 4.3.2. Define $\mathcal{I} = \{(i,j) : 1 \leq i < j \leq n$ and $\mathbf{Y}_{\mathcal{C}_i}, \mathbf{Y}_{\mathcal{C}_j}$ belong to different clusters$\}$, and $\mathcal{J} = \{(i,j) : 1 \leq i < j \leq n$ and $\mathbf{Y}_{\mathcal{C}_i}, \mathbf{Y}_{\mathcal{C}_j}$ belong to the same cluster$\}$. Similar to the proof of Theorem 4.3.1, let $\mathcal{E}_1$ be the event that correct neighborhood property holds for all $\{\mathbf{Y}_{\mathcal{C}_j}\}_{i=j}^n$, let $\mathcal{E}_2$ be the event $\{\sigma||\mathbf{e}_i^{(k)}||_2 < g, \forall i,k\}$ (g is from Assumption $A4$), $\mathcal{E}_3$ is the event that the smallest singular value of $\mathbf{B}\mathbf{B}^T$ is at least q_0, $\forall i = 1, ..., n$, and $\mathcal{E}_4$ is the event that the sub-cluster preserving property is satisfied.

We will show that conditioning on $\mathcal{E}_2$, $\mathcal{E}_3$ and $\mathcal{E}_4$, there is a deterministic upper bound l on $d(\mathbf{Y}_{\mathcal{C}_i}, \mathbf{Y}_{\mathcal{C}_j})_{(i,j)\in\mathcal{J}}$, therefore we have

$$\mathbb{P}\left[\mathcal{E}_1|\mathcal{E}_2,\mathcal{E}_3,\mathcal{E}_4\right] \geq \mathbb{P}\left[d(\mathbf{Y}_{\mathcal{C}_i}, \mathbf{Y}_{\mathcal{C}_j})_{\forall(i,j)\in\mathcal{I}} > l \middle| \mathcal{E}_2,\mathcal{E}_3,\mathcal{E}_4\right] \geq 1 - \sum_{\forall(i,j)\in\mathcal{I}} \mathbb{P}\left[d(\mathbf{Y}_{\mathcal{C}_i}, \mathbf{Y}_{\mathcal{C}_j}) \leq l \middle| \mathcal{E}_2,\mathcal{E}_3,\mathcal{E}_4\right].$$

Then we obtain an upper bound on $\mathbb{P}[d(\mathbf{Y}_{\mathcal{C}_i}, \mathbf{Y}_{\mathcal{C}_k}) \leq l \mid \mathcal{E}_2,\mathcal{E}_3,\mathcal{E}_4]$, $(i,k) \in \mathcal{I}$. The theorem will follow by using the union bound.

WLOG assume that $\mathbf{Y}_{\mathcal{C}_1}$ and $\mathbf{Y}_{\mathcal{C}_2}$ belong to $\mathcal{S}_1$, and $\mathbf{Y}_{\mathcal{C}_3}$ belongs to $\mathcal{S}_2$. The distance function $d(\mathbf{Y}_{\mathcal{C}_1}, \mathbf{Y}_{\mathcal{C}_2})$ can be explicitly written as

$$\left\|\mathbf{Y}_{\mathcal{C}_1} - \mathbf{Y}_{\mathcal{C}_2}(\mathbf{Y}_{\mathcal{C}_2}^T\mathbf{Y}_{\mathcal{C}_2} + \lambda\mathbf{I})^{-1}\mathbf{Y}_{\mathcal{C}_2}^T\mathbf{Y}_{\mathcal{C}_1}\right\|_F + \left\|\mathbf{Y}_{\mathcal{C}_2} - \mathbf{Y}_{\mathcal{C}_1}(\mathbf{Y}_{\mathcal{C}_1}^T\mathbf{Y}_{\mathcal{C}_1} + \lambda\mathbf{I})^{-1}\mathbf{Y}_{\mathcal{C}_1}^T\mathbf{Y}_{\mathcal{C}_2}\right\|_F, \tag{C.18}$$

where $\mathbf{Y}_{\mathcal{C}_1} = \mathbf{U}_1\hat{\mathbf{B}}_1 + \hat{\mathbf{E}}_1$ (similar forms for $\mathbf{Y}_{\mathcal{C}_2}$ and $\mathbf{Y}_{\mathcal{C}_3}$). Using equation (C.17), the first term in (C.18) can be rewritten as

$$\begin{aligned}
&\left\|\mathbf{Y}_{\mathcal{C}_1} - \mathbf{Y}_{\mathcal{C}_2}(\mathbf{Y}_{\mathcal{C}_2}^T\mathbf{Y}_{\mathcal{C}_2} + \lambda\mathbf{I})^{-1}\mathbf{Y}_{\mathcal{C}_2}^T\mathbf{Y}_{\mathcal{C}_1}\right\|_F \\
=&\left\|\mathbf{Y}_{\mathcal{C}_1} - (\mathbf{Y}_{\mathcal{C}_2}\mathbf{Y}_{\mathcal{C}_2}^T + \lambda\mathbf{I} - \lambda\mathbf{I})(\mathbf{Y}_{\mathcal{C}_2}\mathbf{Y}_{\mathcal{C}_2}^T + \lambda\mathbf{I})^{-1}\mathbf{Y}_{\mathcal{C}_1}\right\|_F \\
=&\lambda\left\|(\mathbf{Y}_{\mathcal{C}_2}\mathbf{Y}_{\mathcal{C}_2}^T + \lambda\mathbf{I})^{-1}\mathbf{Y}_{\mathcal{C}_1}\right\|_F \\
<&\lambda\left\|[(\mathbf{Y}_{\mathcal{C}_2}\mathbf{Y}_{\mathcal{C}_2}^T + \lambda\mathbf{I})^{-1} - (\mathbf{U}_1\hat{\mathbf{B}}_2\hat{\mathbf{B}}_2^T\mathbf{U}_1^T + \lambda\mathbf{I})^{-1}]||_F||\mathbf{Y}_{\mathcal{C}_1}\right\|_F \\
&+\lambda\left\|(\mathbf{U}_1\hat{\mathbf{B}}_2\hat{\mathbf{B}}_2^T\mathbf{U}_1^T + \lambda\mathbf{I})^{-1}\mathbf{Y}_{\mathcal{C}_1}\right\|_F \\
<&\lambda\left\|(\mathbf{Y}_{\mathcal{C}_2}\mathbf{Y}_{\mathcal{C}_2}^T + \lambda\mathbf{I})^{-1} - (\mathbf{U}_1\hat{\mathbf{B}}_2\hat{\mathbf{B}}_2^T\mathbf{U}_1^T + \lambda\mathbf{I})^{-1}\right\|_F\sqrt{d_{\max}+1} \\
&+\lambda\left\|(\mathbf{U}_1\hat{\mathbf{B}}_2\hat{\mathbf{B}}_2^T\mathbf{U}_1^T + \lambda\mathbf{I})^{-1}\right\|_F\sqrt{d_{max}+1}.
\end{aligned} \tag{C.19}$$

Now we are going to complete our proof in 3 steps. Unless specified otherwise, the following Step 1 to Step 3 are derived conditioning on $\mathcal{E}_2$, $\mathcal{E}_3$ and $\mathcal{E}_4$.

Step 1: We can rewrite the first term in (C.19) as

$$\lambda\left\|(\mathbf{G}_2 + \mathbf{H})^{-1} - \mathbf{H}^{-1}\right\|_F\sqrt{d_{\max}+1},$$

where $\mathbf{H} = \mathbf{U}_1\hat{\mathbf{B}}_2\hat{\mathbf{B}}_2^T\mathbf{U}_1^T + \lambda\mathbf{I}$, and $\mathbf{G}_2 = \mathbf{Y}_{\mathcal{C}_2}\mathbf{Y}_{\mathcal{C}_2}^T - \mathbf{U}_1\hat{\mathbf{B}}_2\hat{\mathbf{B}}_2^T\mathbf{U}_1^T = \hat{\mathbf{E}}_2\hat{\mathbf{B}}_2^T\mathbf{U}_1^T + \mathbf{U}_1\hat{\mathbf{B}}_2\hat{\mathbf{E}}_2^T + \hat{\mathbf{E}}_2\hat{\mathbf{E}}_2^T$.

Note that the normalizing constant of each column of $\{\hat{\mathbf{E}}_i\}_{i=1}^n$ are bounded in $[1-g, 1+g]$. We then have

$$\begin{aligned}\|\mathbf{G}_2\|_F &\leq \left\|\hat{\mathbf{E}}_2\right\|_F \left\|\hat{\mathbf{B}}_2^T\mathbf{U}_1^T\right\|_F + \left\|\mathbf{U}_1\hat{\mathbf{B}}_2 + \hat{\mathbf{E}}_2\right\|_F \left\|\hat{\mathbf{E}}_2^T\right\|_F \\ &\leq \frac{(2g-g^2)(d_{max}+1)}{(1-g)^2}. \end{aligned} \tag{C.20}$$

The above analysis used triangle inequalities and the bounds of normalizing constants.

Using the fact that $\left\|\mathbf{H}^{-1}\right\|_F < \frac{(1+g)\sqrt{d}}{q_0}$ and inequality (C.20), we have

$$\left\|\mathbf{H}^{-1}\mathbf{G}_2\right\|_F \leq \left\|\mathbf{H}^{-1}\right\|_F \|\mathbf{G}_2\|_F = \frac{(2g-g^2)(1+g)\sqrt{d}\,(d_{max}+1)}{q_0(1-g)}\frac{1}{2} =: f(d) < \frac{1}{2}.$$

Therefore $\lim_{m\to\infty}(\mathbf{H}^{-1}\mathbf{G}_2)^m = \mathbf{0}$. From Theorem 4.29 in Schott [2016] we know $(\mathbf{I}+\mathbf{H}^{-1}\mathbf{G}_2)^{-1} = \sum_{j=0}^{\infty}(\mathbf{H}^{-1}\mathbf{G}_2)^j$ and

$$\begin{aligned}\left\|(\mathbf{G}_2+\mathbf{H})^{-1} - \mathbf{H}^{-1}\right\|_F &= \left\|\mathbf{H}^{-1}\mathbf{G}(\mathbf{I}+\mathbf{H}^{-1}\mathbf{G}_2)^{-1}\mathbf{H}^{-1}\right\|_F \\ &\leq \left\|\sum_{j=1}^{\infty}(\mathbf{H}^{-1}\mathbf{G}_2)^j\right\|_F \left\|\mathbf{H}^{-1}\right\|_F \\ &< \frac{\left\|\mathbf{H}^{-1}\mathbf{G}_2\right\|_F}{1-\|\mathbf{H}^{-1}\mathbf{G}_2\|_F}\frac{\sqrt{d}(1+g)}{q_0} = \frac{(1+g)f(d)\sqrt{d}}{q_0(1-f(d))}.\end{aligned}$$

We then have for the first term in (C.19)

$$\lambda\left\|(\mathbf{G}_2+\mathbf{H})^{-1} - \mathbf{H}^{-1}\right\|_F\sqrt{d_{\max}+1} < \frac{\lambda(1+g)\sqrt{d(d_{max}+1)}f(d)}{q_0\,(1-f(d))}.$$

For the second term in (C.19) we have

$$\lambda\left\|(\mathbf{U}_1\hat{\mathbf{B}}_2\hat{\mathbf{B}}_2^T\mathbf{U}_1^T + \lambda\mathbf{I})^{-1}\right\|_F\sqrt{d_{max}+1} \leq \frac{\lambda(1+g)\sqrt{d(d_{max}+1)}}{q_0}.$$

Hence (C.19) can be upper bounded by $\frac{2\lambda(1+g)\sqrt{d(d_{max}+1)}}{q_0(1-g)}$. Note this quantity is deterministic and does not depend on the choices of $\{\mathbf{B}_i\}_{i=1}^n$ and $\{\mathbf{U}_k\}_{k=1}^K$. The manipulation on RHS of (C.18) is

symmetric, therefore we set

$$l := \frac{4\lambda(1+g)\sqrt{d(d_{max}+1)}}{q_0(1-g)} > d(\mathbf{Y}_{\mathcal{C}_i}, \mathbf{Y}_{\mathcal{C}_j})_{(i,j)\in\mathcal{J}}.$$

Step 2: Now we consider $\mathbb{P}\left[d\left(\mathbf{Y}_{\mathcal{C}_1}, \mathbf{Y}_{\mathcal{C}_3}\right) \leq l | \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4\right]$. We explicitly write $d(\mathbf{Y}_{\mathcal{C}_1}, \mathbf{Y}_{\mathcal{C}_3})$ as

$$\left\|\mathbf{Y}_{\mathcal{C}_1} - \mathbf{Y}_{\mathcal{C}_3}(\mathbf{Y}_{\mathcal{C}_3}^T\mathbf{Y}_{\mathcal{C}_3} + \lambda\mathbf{I})^{-1}\mathbf{Y}_{\mathcal{C}_3}^T\mathbf{Y}_{\mathcal{C}_1}\right\|_F + \left\|\mathbf{Y}_{\mathcal{C}_3} - \mathbf{Y}_{\mathcal{C}_1}(\mathbf{Y}_{\mathcal{C}_1}^T\mathbf{Y}_{\mathcal{C}_1} + \lambda\mathbf{I})^{-1}\mathbf{Y}_{\mathcal{C}_1}^T\mathbf{Y}_{\mathcal{C}_3}\right\|_F. \tag{C.21}$$

Note the following relation

$$\begin{aligned}\mathbb{P}\left[d(\mathbf{Y}_{\mathcal{C}_1}, \mathbf{Y}_{\mathcal{C}_3}) \leq l | \mathcal{E}_2, \mathcal{E}_3\right] \leq & \mathbb{P}\left[\left\|\mathbf{Y}_{\mathcal{C}_1} - \mathbf{Y}_{\mathcal{C}_3}(\mathbf{Y}_{\mathcal{C}_3}^T\mathbf{Y}_{\mathcal{C}_3} + \lambda\mathbf{I})^{-1}\mathbf{Y}_{\mathcal{C}_3}^T\mathbf{Y}_{\mathcal{C}_1}\right\|_F \leq \frac{l}{2}\middle| \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4\right] \\ & + \mathbb{P}\left[\left\|\mathbf{Y}_{\mathcal{C}_3} - \mathbf{Y}_{\mathcal{C}_1}(\mathbf{Y}_{\mathcal{C}_1}^T\mathbf{Y}_{\mathcal{C}_1} + \lambda\mathbf{I})^{-1}\mathbf{Y}_{\mathcal{C}_1}^T\mathbf{Y}_{\mathcal{C}_3}\right\|_F \leq \frac{l}{2}\middle| \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4\right]. \end{aligned} \tag{C.22}$$

To bound the first term in (C.21), we only need to use reverse triangle inequalities and proceed as before. Specifically

$$\begin{aligned} & \left\|\mathbf{Y}_{\mathcal{C}_1} - \mathbf{Y}_{\mathcal{C}_3}\left(\mathbf{Y}_{\mathcal{C}_3}^T\mathbf{Y}_{\mathcal{C}_3} + \lambda\mathbf{I}\right)^{-1}\mathbf{Y}_{\mathcal{C}_3}^T\mathbf{Y}_{\mathcal{C}_1}\right\|_F \\ = & \lambda\left\|\left(\mathbf{Y}_{\mathcal{C}_3}\mathbf{Y}_{\mathcal{C}_3}^T + \lambda\mathbf{I}\right)^{-1}\mathbf{Y}_{\mathcal{C}_1}\right\|_F \\ > & \lambda\left\|\left(\mathbf{U}_2\hat{\mathbf{B}}_3\hat{\mathbf{B}}_3^T\mathbf{U}_2^T + \lambda\mathbf{I}\right)^{-1}\mathbf{U}_1\hat{\mathbf{B}}_1\right\|_F - \lambda\left\|\left(\mathbf{U}_2\hat{\mathbf{B}}_3\hat{\mathbf{B}}_3^T\mathbf{U}_2^T + \lambda\mathbf{I}\right)^{-1}\right\|_F\left\|\hat{\mathbf{E}}_1\right\|_F \\ & -\lambda\left\|\left[\left(\mathbf{Y}_{\mathcal{C}_3}\mathbf{Y}_{\mathcal{C}_3}^T + \lambda\mathbf{I}\right)^{-1} - \left(\mathbf{U}_2\hat{\mathbf{B}}_3\hat{\mathbf{B}}_3^T\mathbf{U}_2^T + \lambda\mathbf{I}\right)^{-1}\right]\right\|_F\sqrt{d_{\max}+1}. \end{aligned}$$

The last two terms are upper bounded by $\frac{\lambda(1+g)\sqrt{d(d_{max}+1)}}{q_0(1-g)}$ as before. For the first term

$$\begin{aligned} & \lambda\left\|\left(\mathbf{U}_2\hat{\mathbf{B}}_3\hat{\mathbf{B}}_3^T\mathbf{U}_2^T + \lambda\mathbf{I}\right)^{-1}\mathbf{U}_1\hat{\mathbf{B}}_1\right\|_F \\ \geq & \left\|\mathbf{U}_1\hat{\mathbf{B}}_1 - \mathbf{U}_2\mathbf{U}_2^T\mathbf{U}_1\hat{\mathbf{B}}_1\right\|_F - \lambda\left\|\mathbf{U}_2\left(\hat{\mathbf{B}}_3\hat{\mathbf{B}}_3^T + \lambda\mathbf{I}\right)^{-1}\mathbf{U}_2^T\mathbf{U}_1\hat{\mathbf{B}}_1\right\|_F \\ > & \left\|\mathbf{U}_1\hat{\mathbf{B}}_1 - \mathbf{U}_2\mathbf{U}_2^T\mathbf{U}_1\hat{\mathbf{B}}_1\right\|_F - \frac{\lambda(1+g)\sqrt{d(d_{max}+1)}}{q_0(1-g)}, \end{aligned} \tag{C.23}$$

where inequality (C.23) comes from the following relations

$$\lambda \left\| \mathbf{U}_2 \left(\hat{\mathbf{B}}_3 \hat{\mathbf{B}}_3^T + \lambda \mathbf{I} \right)^{-1} \mathbf{U}_2^T \mathbf{U}_1 \hat{\mathbf{B}}_1 \right\|_F \leq \lambda \left\| \left(\hat{\mathbf{B}}_3 \hat{\mathbf{B}}_3^T + \lambda \mathbf{I} \right)^{-1} \right\|_F \left\| \mathbf{U}_1 \hat{\mathbf{B}}_1 \right\|_F$$
$$\leq \lambda \frac{\sqrt{d}(1+g)}{q_0} \frac{\sqrt{d_{max}+1}}{1-g} = \frac{\lambda(1+g)\sqrt{d(d_{max}+1)}}{q_0(1-g)}.$$

For the first term in (C.23) we have

$$\left\| \mathbf{U}_1 \hat{\mathbf{B}}_1 - \mathbf{U}_2 \mathbf{U}_2^T \mathbf{U}_1 \hat{\mathbf{B}}_1 \right\|_F = \sqrt{Tr \left[\hat{\mathbf{B}}_1^T \hat{\mathbf{B}}_1 - \hat{\mathbf{B}}_1^T \mathbf{U}_1^T \mathbf{U}_2 \mathbf{U}_2^T \mathbf{U}_1 \hat{\mathbf{B}}_1 \right]}$$
$$= \left\| \sqrt{\mathbf{I} - \mathbf{\Lambda}_{12}^2} \tilde{\mathbf{B}}_1 \mathbf{W} \right\|_F \geq \frac{\left\| \sqrt{\mathbf{I} - \mathbf{\Lambda}_{12}^2} \tilde{\mathbf{B}}_1 \right\|_F}{1+g},$$

where $\mathbf{W}$ is the diagonal matrix with its diagonal equals to the reciprocal of normalizing constants of each column of $\hat{\mathbf{B}}_1$ (simply note $\hat{\mathbf{B}}_1 = \mathbf{B}_1 \mathbf{W}$), $\tilde{\mathbf{B}}_1 = \mathbf{V}\mathbf{B}_1$ is a orthogonal transformation of $\mathbf{B}_1$ (here $\mathbf{V}$ is the right orthogonal matrix in the svd of $\mathbf{U}_2^T \mathbf{U}_1$), and $\mathbf{\Lambda}_{12}$ is the diagonal matrix that takes λ_i^{12} $(i = 1, .., d)$ as its i-th diagonal entry. Therefore, eventually the first term at the RHS of (C.22) can be upper bounded by

$$\mathbb{P} \left[\left\| \sqrt{\mathbf{I} - \mathbf{\Lambda}_{12}^2} \tilde{\mathbf{B}}_1 \right\|_F \leq \frac{5\lambda(1+g)^2 \sqrt{d(d_{max}+1)}}{q_0(1-g)} \right].$$

Using Assumption A4, Lemma C.1.1 and arguments similar to the proof of Theorem 4.3.1, we know the quantity above is upper bounded by

$$\mathbb{P} \left[\left\| \sqrt{\mathbf{I} - \mathbf{\Lambda}_{12}^2} \tilde{\mathbf{B}}_{1'1} \right\|_F \leq \sqrt{1 - T_l^2} \right] \leq 2e^{-\epsilon^2},$$

where $\tilde{\mathbf{B}}_{1'1}$ is the first column of $\tilde{\mathbf{B}}_1$, and ϵ is the same as in Theorem 4.3.1. Using analogous manipulations we obtain similar results for the second term in (C.22). Therefore $\mathbb{P}\left[d(\mathbf{Y}_{\mathcal{C}_1}, \mathbf{Y}_{\mathcal{C}_3}) \leq l | \mathcal{E}_2\right] \leq 4e^{-\epsilon^2}$.

Step 3: Now we are going to lower bound $\mathbb{P}\left[\mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4\right]$. Just as in the proof of Theorem 4.3.1 we have

$$\mathbb{P} \left[\sigma \mathbf{e}_i^{(k)} \geq g \right] \leq 2e^{-t^2},$$

where $t = \frac{D(\frac{g^2}{D\sigma^2}-1)}{2(\sqrt{D}+\frac{Dg^2}{\sigma^2\sqrt{d}}+\sqrt{d})}$. Hence from Assumption $A4$ we know $2e^{-t^2} \leq \frac{2}{N^{(1+\frac{\eta}{2+\eta})^2}}$. Using union bound inequality we have

$$\mathbb{P}\left[\mathcal{E}_2\right] \geq 1 - \frac{2N}{N^{(1+\frac{\eta}{2+\eta})^2} - 2}. \tag{C.24}$$

From Lemma C.1.5 we have

$$\mathbb{P}\left[\mathcal{E}_3\right] \geq 1 - \frac{2n}{N^{t^2/2}} - \sum_{k=1}^{K} \frac{2n_k(N_k - d_{max})}{d_{max}(N_k+1)(N_k^{\rho}-1)^2}. \tag{C.25}$$

From the assumption we have

$$\mathbb{P}\left[\mathcal{E}_4\right] \geq 1 - p_s. \tag{C.26}$$

Combing (C.24), (C.25) and (C.26) we know

$$\begin{aligned}
\mathbb{P}[\mathcal{E}_1] &\geq 1 - \sum_{\forall(i,j)\in\mathcal{I}} \mathbb{P}\left[d(\mathbf{Y}_{\mathcal{C}_i}, \mathbf{Y}_{\mathcal{C}_j}) \leq l \middle| \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4\right] - \mathbb{P}\left[\mathcal{E}_2^{\complement}\right] - \mathbb{P}\left[\mathcal{E}_3^{\complement}\right] - \mathbb{P}\left[\mathcal{E}_4^{\complement}\right] \\
&\geq 1 - p_s - 4n(n-1)e^{-\epsilon^2} - \frac{2n}{N^{t^2/2}} - \sum_{k=1}^{K} \frac{2n_k(N_k - d_{max})}{d_{max}(N_k+1)(N_k^{\rho}-1)^2} - \frac{2N}{N^{(1+\frac{\eta}{2+\eta})^2} - 2}.
\end{aligned} \tag{C.27}$$

This completes the proof. □

C.2 Residual Minimization by Ridge Regression

In this section we provide the algorithm for classifying the out-of-sample points.

input : $\mathbf{Y}$ to be classified, $\mathbf{R}$ and $\boldsymbol{\ell}$ are the training data and labels, m and λ are the residual minimization and regularization parameters
output : The label vector $\boldsymbol{\ell}$ of all points in $\mathbf{Y}$
1. Generate subsets of training data
for $k = 1$ **to** K **do**
  Uniformly sample m points from the k-th cluster in the training set $\mathbf{R}$, denote this sampled set as $\mathbf{R}_k$;
end
2. Compute the projection matrix for each cluster
for $k = 1$ **to** K **do**
  $\mathbf{P}_k := \mathbf{R}_k(\mathbf{R}_k^T\mathbf{R}_k + \lambda\mathbf{I})^{-1}\mathbf{R}_k^T$
end
3. Compute residuals for points in Y, here N is the number of points in $\mathbf{Y}$
for $i = 1$ **to** N **do**
  for $k = 1$ **to** K **do**
    $r_i(k) := (\mathbf{I} - \mathbf{P}_k)\mathbf{y}_i$;
  end
end
4. Assign labels through minimum residual
for $i = 1$ **to** N **do**
  $\ell_i = \arg\min_k \{r_i(k)\}$;
end

Algorithm 4: Residual Minimization by Ridge Regression (RMRR) algorithm.

C.3 Additional Numerical Results

In this section, we present additional numerical results. Results for some algorithms are omitted for certain datasets due to the limitations on computational resources.

C.3.1 Results on Extended Yale B

Method	Accuracy (%)	Accuracy-Sub (%)	NMI (%)	Runtime (sec.)
SBSC-SSC	20.99 (1.02)	22.5 (1.29)	34.24 (1.15)	56
SSSC	49.33 (2.51)	56.54 (1,77)	52.82 (2.21)	22.11471427
LRR	55.63	NA	64.02	28.68
LSR	54.11	NA	65.12	7.56

Table C.1: Additional results on Extended Yale B

C.3.2 Results on Zipcode

Method	Accuracy (%)	Accuracy-Sub (%)	NMI (%)	Runtime (sec.)
SBSC-TSC(6)	73 (3.96)	72.33 (2.02)	76.17 (1.76)	66.46
SBSC-DSC(6)	66.42 (4.91)	66.52 (2.43)	71.28 (1.24)	413.98
SBSC-SSC(1)	55.24 (1.34)	61.44 (2.36)	45.18 (1.42)	116.72
SBSC-SSC(6)	65 (0.44)	61.1 (0.79)	62.88 (0.88)	674.14
STSC(6)	57.76 (1.15)	60.1 (1.8)	60.4 (1.59)	13.12
SDSC(6)	49.18 (2.19)	50.28 (1.39)	53.37 (1.5)	19.47

SSSC(1)	41.52 (5.92)	44.86 (7.06)	38.22 (3.7)	24.88
SSSC(6)	44.43 (4)	44.06 (2.53)	42.61 (2.23)	132.84
SLRR(6)	63.7 (3.74)	63.85 (1.74)	69.25 (1.86)	45.87
SLSR(6)	60.71 (1.04)	59.43 (0.8)	66.39 (1.08)	25.62
LRR	53.25	NA	53.53	401
LSR	58.91	NA	61.56	191.5

Table C.2: Additional results on Zipcode

C.3.3 Results on MNIST

Method	**Accuracy (%)**	**Accuracy-Sub (%)**	**NMI (%)**	**Runtime (sec.)**
SBSC-SSC	84.95 (4.51)	86.48 (4.2)	73.71 (2.06)	834.42
SSSC(1)	33.26 (2.15)	77.22 (3.9)	13.59 (1.41)	43.21
SSSC(6)	48.49 (2.75)	79.06 (1.63)	30.41 (2.04)	259

Table C.3: Additional results on MNIST

C.4 Additional Technical Discussions

C.4.1 The ϵ in Theorem 4.3.1

In this section, we will show that under mild conditions, ϵ in (4.9) is $O(\sqrt{d})$. For ease of notation, we write $r_i = \left(g_1^2 - \lambda_i^{(1)2}\right)_+$ and $s_i = \left(g_1^2 - \lambda_i^{(1)2}\right)_-$. WLOG assume ϵ is evaluated at $k = 1$.

Main result: If there exist constants $c_1 \in (0, g_1]$, $c_2 \in (0, 1)$ and $c_3 > 0$ such that $\sum_{i=1}^d r_i \geq c_1 d$, $\frac{\sum_{i=1}^d s_i}{\sum_{i=1}^d r_i} \leq c_2$ and $c_3 d > g_1$, then we have

$$\epsilon \geq \frac{(1-c_2)\sqrt{d}}{2\sqrt{\frac{(c_1+c_3)g_1}{c_1^2}} + \sqrt{\frac{4(c_1+c_3)g_1}{c_1^2} + \frac{2}{c_1}}}.$$

Proof. Note that

$$\epsilon = \frac{1 - \frac{\sum_{i=1}^d s_i}{\sum_{i=1}^d r_i}}{2\sqrt{\frac{\sum_{i=1}^d r_i^2}{(\sum_{i=1}^d r_i)^2}} + \sqrt{\frac{4\sum_{i=1}^d r_i^2}{(\sum_{i=1}^d r_i)^2} + \frac{2}{\sum_{i=1}^d r_i}}}. \tag{C.28}$$

Define $f : \mathcal{V} \to R$, where $f(\mathbf{v}) = \frac{\sum_{i=1}^d v_i^2}{(\sum_{i=1}^d v_i)^2}$, and $\mathcal{V} = \{\mathbf{v} \in [0, g_1]^d : \sum_{i=1}^d v_i = \sum_{i=1}^d r_i\}$. Consider the following $\mathbf{r}^* \in \mathcal{V}$

$$r_i^* = \begin{cases} g_1, & \text{if } i \leq \lfloor \frac{\sum_{i=1}^d r_i}{g_1} \rfloor, \\ \sum_{i=1}^d r_i - \lfloor \frac{\sum_{i=1}^d r_i}{g_1} \rfloor \cdot g_1, & i = \lfloor \frac{\sum_{i=1}^d r_i}{g_1} \rfloor + 1, \\ 0, & \text{otherwise.} \end{cases}$$

We will prove by contradiction that any maximizer of $f(\cdot)$ is a permutation of $\mathbf{r}^*$.

In fact, assume $\mathbf{r}' \in \mathcal{V}$ also maximizes $f(\cdot)$ but is not a permutation of $\mathbf{r}^*$. Assume there are m terms in $\{r_i'\}_{i=1}^d$ that are equal to g_1. Let $r_1' \leq r_2'$ be the two smallest positive terms of $\{r_i'\}_{i=1}^d$. It is straightforward to see $r_2' < g_1$. Consequently, we can find a constant $\delta > 0$ such that $r_1' - \delta, r_2' + \delta \in (0, g_1)$. Note $\mathbf{r}'' = [r_1' - \delta, r_2' + \delta, r_3', ..., r_d'] \in \mathcal{V}$, but $f(\mathbf{r}'') > f(\mathbf{r}')$, which is a contradiction.

Note that $\mathbf{r} \in \mathcal{V}$, we plug $\mathbf{r}^*$ into $f(\cdot)$ and get

$$f(\mathbf{r}) = \frac{\sum_{i=1}^d r_i^2}{(\sum_{i=1}^d r_i)^2} \leq \frac{(\frac{\sum_{i=1}^d r_i}{g_1} + 1)g_1^2}{(\sum_{i=1}^d r_i)^2} \leq \frac{(\frac{c_1 d}{g_1} + 1)g_1^2}{(c_1 d)^2} \leq \frac{(c_1 + c_3)g_1}{c_1^2 d}.$$

Finally, from the inequality above and (C.28) we have

$$\epsilon \geq \frac{1 - c_2}{2\sqrt{\frac{(c_1+c_3)g_1}{c_1^2 d}} + \sqrt{\frac{4(c_1+c_3)g_1}{c_1^2 d} + \frac{2}{c_1 d}}} = \frac{(1 - c_2)\sqrt{d}}{2\sqrt{\frac{(c_1+c_3)g_1}{c_1^2}} + \sqrt{\frac{4(c_1+c_3)g_1}{c_1^2} + \frac{2}{c_1}}}.$$

□

BIBLIOGRAPHY

M. Anantharaman, S. Sindhu, S. Jagatheesan, K. Malini, and P. Kurian. Dielectric properties of rubber ferrite composites containing mixed ferrites. *Journal of Physics D: Applied Physics*, (15): 1801, 1999.

T. S. Anantharaman, B. Mishra, and D. C. Schwartz. Genomics via optical mapping ii: Ordered restriction maps. *Journal of Computational Biology*, 4(2):91–118, 1997.

P. A. Audano, A. Sulovari, T. A. Graves-Lindsay, S. Cantsilieris, M. Sorensen, A. E. Welch, M. L. Dougherty, B. J. Nelson, A. Shah, S. K. Dutcher, et al. Characterizing the major structural variant alleles of the human genome. *Cell*, 176(3):663–675, 2019.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

R. B. Cattell. Factor analysis: an introduction and manual for the psychologist and social scientist. 1952.

M. J. Chaisson, R. K. Wilson, and E. E. Eichler. Genetic variation and the de novo assembly of human genomes. *Nature Reviews Genetics*, 16(11):627, 2015.

V. Chvátal. The tail of the hypergeometric distribution. *Discrete Mathematics*, 25(3):285–287, 1979.

E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk. Greedy feature selection for subspace clustering. *The Journal of Machine Learning Research*, 14(1):2487–2517, 2013.

E. Elhamifar and R. Vidal. Sparse subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.

X. Fan, J. Xu, and L. Nakhleh. Detecting large indels using optical map data. In *RECOMB International conference on Comparative Genomics*, pages 108–127. Springer, 2018.

R. Heckel and H. Bölcskei. Robust subspace clustering via thresholding. *IEEE Transactions on Information Theory*, 61(11):6320–6342, 2015.

W. Hong, J. Wright, K. Huang, and Y. Ma. Multiscale hybrid linear models for lossy image representation. *IEEE Transactions on Image Processing*, 15(12):3655–3671, 2006.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

K. Howe and J. M. Wood. Using optical mapping data for the improvement of vertebrate genome assemblies. *GigaScience*, 4(1):s13742–015, 2015.

J. Huddleston and E. E. Eichler. An incomplete understanding of human genetic variation. *Genetics*, 202(4):1251–1254, 2016.

M. E. Hurles, E. T. Dermitzakis, and C. Tyler-Smith. The functional impact of structural variation in humans. *Trends in Genetics*, 24(5):238–245, 2008.

B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

G. Lerman, M. McCoy, J. A. Tropp, and T. Zhang. Robust computation of linear models, or how to find a needle in a haystack. Technical report, CALIFORNIA INST OF TECH PASADENA DEPT OF COMPUTING AND MATHEMATICAL SCIENCES, 2012.

A. K.-Y. Leung, N. Jin, K. Y. Yip, and T.-F. Chan. Omtools: a software package for visualizing and processing optical mapping data. *Bioinformatics*, 33(18):2933–2935, 2017a.

A. K.-Y. Leung, T.-P. Kwok, R. Wan, M. Xiao, P.-Y. Kwok, K. Y. Yip, and T.-F. Chan. Omblast: alignment tool for optical mapping using a seed-and-extend approach. *Bioinformatics*, 33(3): 311–319, 2017b.

J. Lin. *On the dirichlet distribution*. PhD thesis, 2016.

G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 663–670, 2010.

R. Liu, R. Hao, and Z. Su. Mixture of manifolds clustering via low rank embedding. *Journal of Information and Computational Science*, 8:725–737, 2011.

C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. In *European conference on computer vision*, pages 347–360. Springer, 2012.

J. Luebeck, C. Coruh, S. R. Dehkordi, J. T. Lange, K. M. Turner, V. Deshpande, D. A. Pai, C. Zhang, U. Rajkumar, J. A. Law, et al. Ampliconreconstructor: Integrated analysis of ngs and optical mapping resolves the complex structures of focal amplifications in cancer. *bioRxiv*, 2020.

U. V. Luxburg, O. Bousquet, and M. Belkin. Limits of spectral clustering. In *Advances in neural information processing systems*, pages 857–864, 2005.

M. D. Muggli, S. J. Puglisi, and C. Boucher. Efficient indexed alignment of contigs to optical maps. In *International Workshop on Algorithms in Bioinformatics*, pages 68–81. Springer, 2014.

J. C. Mullikin and Z. Ning. The phusion assembler. *Genome research*, 13(1):81–90, 2003.

K. P. Murphy. Machine learning: a probabilistic perspective. 2012.

N. Nagarajan, T. D. Read, and M. Pop. Scaffolding and validation of bacterial genome assemblies using optical restriction maps. *Bioinformatics*, 24(10):1229–1235, 2008.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

D. Park, C. Caramanis, and S. Sanghavi. Greedy subspace clustering. In *Advances in Neural Information Processing Systems*, pages 2753–2761, 2014.

M. Pendleton, R. Sebra, A. W. C. Pang, A. Ummat, O. Franzen, T. Rausch, A. M. Stütz, W. Stedman, T. Anantharaman, A. Hastie, et al. Assembly and diploid architecture of an individual human genome via single-molecule technologies. *Nature methods*, 12(8):780, 2015.

X. Peng, L. Zhang, and Z. Yi. Scalable sparse subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 430–437. IEEE, 2013.

X. Peng, H. Tang, L. Zhang, Z. Yi, and S. Xiao. A unified framework for representation-based subspace clustering of out-of-sample and large-scale data. *IEEE Transactions on Neural Networks and Learning Systems*, 27(12):2499–2512, 2016. ISSN 2162-237X. doi: 10.1109/TNNLS.2015.2490080.

M. Rahmani and G. K. Atia. Subspace clustering via optimal direction search. *IEEE Signal Processing Letters*, 24(12):1793–1797, 2017.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

J. R. Schott. *Matrix analysis for statistics*. John Wiley & Sons, 2016.

J. M. Shelton, M. C. Coleman, N. Herndon, N. Lu, E. T. Lam, T. Anantharaman, P. Sheth, and S. J. Brown. Tools and pipelines for bionano data: molecule assembly pipeline and fasta super scaffolding tool. *BMC genomics*, 16(1):734, 2015.

L. Shi, Y. Guo, C. Dong, J. Huddleston, H. Yang, X. Han, A. Fu, Q. Li, N. Li, S. Gong, et al. Long-read sequencing and de novo assembly of a chinese genome. *Nature communications*, 7:12065, 2016.

M. Short. Improved inequalities for the poisson and binomial distribution and upper tail quantile functions. *ISRN Probability and Statistics*, 2013.

M. Skala. Hypergeometric tail inequalities: ending the insanity. *arXiv preprint arXiv:1311.5939*, 2013.

B. S. Thomas, L. Lin, L.-H. Lim, and S. Mukherjee. Learning subspaces of different dimension. *arXiv preprint arXiv:1404.6841*, 2014.

J. A. Udall and R. K. Dawe. Is it ordered correctly? validating genome assemblies by optical mapping. *The Plant Cell*, 30(1):7–14, 2018.

A. Valouev, L. Li, Y.-C. Liu, D. C. Schwartz, Y. Yang, Y. Zhang, and M. S. Waterman. Alignment of optical maps. *Journal of Computational Biology*, 13(2):442–462, 2006a.

A. Valouev, D. C. Schwartz, S. Zhou, and M. S. Waterman. An algorithm for assembly of ordered restriction maps from single dna molecules. *Proceedings of the National Academy of Sciences*, 103 (43):15770–15775, 2006b.

A. Valouev, Y. Zhang, D. C. Schwartz, and M. S. Waterman. Refinement of optical map assemblies. *Bioinformatics*, 22(10):1217–1224, 2006c.

R. Vidal. A tutorial on subspace clustering. 28, 01 2010.

R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). *IEEE transactions on pattern analysis and machine intelligence*, 27(12):1945–1959, 2005.

U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

C. You, C. Donnat, D. P. Robinson, and R. Vidal. A divide-and-conquer framework for large-scale subspace clustering. In *Signals, Systems and Computers, 2016 50th Asilomar Conference on*, pages 1014–1018. IEEE, 2016a.

C. You, C.-G. Li, D. P. Robinson, and R. Vidal. Oracle based active set algorithm for scalable elastic net subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3928–3937. IEEE, 2016b.

C. You, D. Robinson, and R. Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3918–3927. IEEE, 2016c.

J. Zhang. *Time Series Modeling with Shape Constraints*. PhD thesis, Columbia University, 2017.

P. Zhou, Y. Hou, and J. Feng. Deep adversarial subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1596–1604. IEEE, 2018.